FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

# ArgMine: Argumentation Mining from Text

**Gil Filipe da Rocha**


Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Henrique Daniel de Avelar Lopes Cardoso (Ph.D.)

Co-Supervisor: Jorge Filipe Pinheiro Guerra de Ribeiro Teixeira

July 28, 2016

# ArgMine:
# Argumentation Mining from Text

**Gil Filipe da Rocha**

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Doctor Ana Paula Cunha da Rocha
External Examiner: Doctor Bruno Emanuel da Graça Martins
Supervisor: Doctor Henrique Daniel de Avelar Lopes Cardoso
Co-Supervisor: Jorge Filipe Pinheiro Guerra de Ribeiro Teixeira

July 28, 2016

# Abstract

The aim of argumentation mining is the automatic detection and identification of the argumentative structure contained within a piece of natural language text. An argument is an ancient and well studied rhetorical structure. In a general form, arguments are justifiable positions where pieces of evidence (premises) are offered in support of a conclusion.

The ambiguity of natural language text, different writing styles, implicit context and the complexity of building argument structures are some of the challenges which make argumentation mining very challenging.

By automatically extracting arguments from text, we are able to tell not just what views are being expressed, but also what are the reasons to believe those particular views. Therefore, argumentation mining has the potential to improve some research topics such as opinion mining, recommender systems and multi-agent systems.

In this thesis the ArgMine Framework is presented, which aims to integrate the creation of an annotated corpus with arguments and the semi-automated process of selection and experimentation of different models and relevant features in different subtasks of the argumentation mining process.

The full task of argumentation mining can be decomposed into several subtasks. This thesis focuses on the automatic detection and identification of the argumentative components presented in the original text, using supervised and semi-supervised machine learning algorithms. The target corpus used to train the supervised machine learning algorithms was manually annotated and is composed of Portuguese news articles, to which argumentation mining does not seem to have been applied before.

The predictive capabilities of the models developed to address the first two subtasks of the argumentation mining process are being exploited to suggest, in unannotated texts, potential arguments to users in the annotation platform.

# Resumo

O objetivo da prospeção de argumentos a partir de texto é a deteção e identificação de forma automática da estrutura argumentativa contida num texto escrito em linguagem natural. Um argumento é uma estrutura retórica que é estudada desde à muitos anos e que se encontra bem fundamentada. De uma forma geral, argumentos são posições justificáveis onde factos (premissas) são apresentados em suporte de uma conclusão.

A ambiguidade do texto escrito em linguagem natural, diferentes estilos de escrita, contexto implícito e a complexidade em construir estruturas argumentativas são alguns dos desafios que fazem desta tarefa muito desafiadora.

Extraindo de forma automática argumentos a partir de texto, somos capazes de saber não apenas quais são os pontos de vista que estão a ser expressos, mas também quais são as razões para acreditar nesses pontos de vista. Assim sendo, a prospeção de argumentos de forma automática tem o potencial de trazer avanços em algumas áreas de investigação tais como prospeção de opiniões, sistemas de recomendação e sistemas multi-agente.

Nesta tese apresentamos a *ArgMine Framework*, a qual tem como objetivo de integrar a criação de um conjunto de dados anotados com argumentos e o processo semi-automático de seleção e experimentação de diferentes modelos e *features* relevantes em diferentes sub-tarefas do processo de prospeção de argumentos.

A tarefa de prospeção de argumentos pode ser decomposta em várias sub-tarefas. Esta tese aborda a deteção e identificação, de forma automática, dos componentes argumentativos presentes no texto, usando algoritmos de aprendizagem máquina supervisionada e semi-supervisionada. O conjunto de dados alvo que será usado para treinar os algoritmos de aprendizagem máquina supervisionada foram manualmente anotados e são constituídos por notícias escritas na língua Portuguesa, na qual a prospeção de argumentos não parece ter sido ainda explorada.

As capacidades preditivas dos modelos desenvolvidos para abordar as primeiras duas subtarefas do processo de prospeção de argumentos estão a ser aplicadas para sugerir, em textos não anotados, potenciais argumentos a utilizadores na plataforma de anotação.

# Acknowledgements

First of all, I would like to show my gratitude to my supervisors, Professor Henrique Lopes Cardoso and Jorge Teixeira.
I would also like to thank my family and friends for all the support until this moment of my life.
Finally, I would like to thank my parents for supporting me all the time and for supporting me in my academic pursuits.

Gil Filipe da Rocha

*"Give a man a fish, and you feed him for a day.*
*Teach a man to fish, and you feed him for a lifetime."*

Chinese Proverb

# Contents

# List of Figures

LIST OF FIGURES

# List of Tables

# Abbreviations

| | |
|---|---|
| ADU | Argumentative Discourse Unit |
| EDU | Elementary Discourse Unit |
| SVM | Support Vector Machine |
| AIF | Argument Interchange Format |
| JSON | JavaScript Object Notation |
| OVA | Online Visualization of Argument |

# Chapter 1

# Introduction

Argumentation is the process whereby arguments are constructed, presented and evaluated. An argument is composed by a set of propositions, where some of them (the premises) are pieces of evidence offered in support of a conclusion. The conclusion is a proposition that has truth-value (which is either true or false), put forward by somebody as true on the basis of the premises.

The ability to engage in the process of argumentation is essential for human beings. Humans use argumentation to communicate and defend their justifiable positions (or opinions), to understand new problems and to perform scientific reasoning. The process of argumentation is commonly used in several areas and plays an important role in some activities. Many professionals (e.g. scientists, lawyers, journalists, politicians) implicitly or explicitly use argumentation in their daily activity. In these situations, argumentation can be used in different phases of the decision process: to analyze a problem or situation, to identify the pros and cons and, to make a decision.

Verbal communication and written texts are the means by which humans communicate their arguments. Thus, we can find arguments almost everywhere: scientific texts, legal texts and court decisions, biomedical texts, patents, reviews, debates, dialogs, news, and so on.

The aim of *argumentation mining* from text, a sub-domain of text mining, is the automatic detection and identification of the argumentative structure contained within a piece of natural language text. As input, this process receives a piece of natural language text. If the text under analysis contains argumentative content, we aim to detect all the arguments that are present in the text document, the relations between them and the internal structure of each individual argument. In the end, this process should be able to output the corresponding *argument diagram*: the visual representation of the arguments presented in the text. An argument diagram is a compact and intuitive representation of the arguments presented in the text, which helps in the process of understanding the points of view that are being expressed, and more interestingly, what are the reasons to believe those particular views.

When argumentation mining is applied on free text, some characteristics of natural language text and from the argumentation process itself make argumentation mining a very challenging task.

The ambiguity of natural language text, different writing styles, implicit context and the complexity of building argument structures from free text are some of the main challenges. As an example, writers typically do not follow the strict rules of grammar because their human readers posses cognitive abilities to interpret the semantic and contextual meaning of the text. As a consequence, when a logic reasoning is being expressed, some parts of the process can be omitted from the text, making the automatic detection of this reasoning process much more challenging.

In order to successfully approach argumentation mining from text, knowledge from different research areas, such as text mining and argumentation theory, will be required. In this thesis, some of the essential concepts and theories of argumentation will be presented. This basic background on argumentation theory will be used throughout this thesis to properly formulate and describe some of the sub-tasks that will be addressed.

## 1.1 Motivation

The skill of distinguishing argument from non-argument spans of text and, the skill of extracting the argumentative structure from free text are sophisticated and require some training. These skills are a typical learning outcome of an undergraduate course on critical thinking that are commonly taught in some graduation degrees, such as philosophy. In this kind of courses, one of the main objectives is to teach students how to interpret and analyze text documents correctly. One of the most useful skills to achieve these goals, is the ability to extract arguments contained within the text and represent them in a intuitive and systematic way. For instance, the ability to properly represent the arguments contained in a piece of text in the form of argument diagrams. In this sense, investigating at which level we can perform this task using a machine learning approach comparing to the human performance is a very interesting question to explore. Also, from the analysis of the results obtained and from the conclusions to be drawn, it will be interesting to discover what patterns that are present in free text are most relevant for the task of argumentation mining.

By automatically extracting arguments from text, we are able to tell not just what views are being expressed, but also what are the reasons to believe those particular views. This is potentially relevant to any kind of text mining application that is directed to argumentative text. Therefore, interesting practical applications become visible in the horizon, such as:

- Legal cases and court decisions: Palau *et al.* [PM09] worked with texts in the legal domain with the aim of automatically detecting and identifying arguments that are presented by the parties involved (e.g., in a court trial), which may significantly enrich the information retrieval capabilities on legal databases, as well as help professionals to evaluate and analyze arguments in the legal domain;

- Scientific text: In the scientific community, every accepted proposition must be supported by facts that can be proven. Therefore, the scientific domain is argumentative in nature.

For instance, in the biomedical domain it is common to find documents presenting supportive evidences when trying to prove that a new experiment is more successful or relevant than another one. The extraction and evaluation of arguments from scientific text can have interesting applications in the process of scientific reasoning;

- User-generated content: argumentation mining can be seen as a natural extension of opinion mining. The aim of opinion mining is to detect user's appreciations or disappointments in relation to something (products or services, for instance). A natural extension is to automatically find the reasons that users provide to justify their point of view [SW12]. Another research community that can benefit from advancements in argumentation mining research is the recommendation systems community. Typically, recommendations presented by recommender systems are based on the classifications that users give to some products or services (e.g., using a 5-star classification). If we are able to know the reasons behind those classifications, we could build systems that better understand user preferences and, therefore, would be able to give better recommendations [CMG09];

- Debates: Discussions of political issues, newspaper articles and opinion articles, are some of the means where we can find relevant arguments for the general public, where tools that help us to visualize, to understand and to evaluate arguments can have interesting applications. Some researchers have explored this direction, including [GLPK14, BD10];

- Education: essays and exams are some of the documents that are written by students during their academic activities, which constitute educational data that can be mined for purposes of assessment and instruction. Other possible applications are: pedagogical tool for argumentation courses, computer-supported peer review, computerized essay grading and large-scale online courses [AS11];

- Multi agent systems: reasoning agents need to communicate with each other and apply argumentation-based reasoning mechanisms to undertake the conflicts arising from their different views of goals, beliefs, and actions [PM09]. Acquisition of knowledge in form of arguments from external resources is also a possible application to this research area.

Preliminary research performed so far yield interesting results, motivating the exploration of more applications in the field. However, before the rise of such applications, the state-of-the-art on argumentation mining must be improved.

Research in the area of argumentation mining is very recent. Initial studies started to appear only a few years ago and within specific domains such as legal texts, online reviews, and debates [Sai12, MP11, CV12]. The growing interest in the topic is tangible. Only in the year 2014, at least three international events occured: the first *ACL* workshop on the topic in Baltimore[1] and meetings dedicated to the topic in both Warsaw[2] and Dundee[3]).

[1] http://www.uncg.edu/cmp/ArgMining2014/
[2] http://argdiap.pl/argdiap2014
[3] http://www.arg-tech.org/swam2014/

Argumentation Mining is a growing research topic that spans across different research areas. Some characteristics of natural language text and from the argumentation process make argumentation mining a very challenging task and, therefore, an engaging problem. In addition, argumentation mining is a research topic with potential for several applications.

## 1.2 Research Goals

The fundamental research questions that will be addressed in this thesis are the following:

- *Given a text document, how to automatically segment the text into argumentative zones?*
- *How can the Argumentative Discourse Units (ADUs) boundaries be automatically identified in free text?*
- *Is it possible to learn, from an annotated corpora, how to identify and extract arguments?*

In this thesis, we aim to work on argumentation mining from text written in the Portuguese language using supervised machine learning algorithms to automatically address some of the sub-steps of the complete argumentation mining process. To achieve this goal using supervised machine learning algorithms, a set of labeled data (corpus) is required. To the best of our knowledge, no such corpus exists. In this thesis, the creation of a corpus with arguments annotated from text written in the Portuguese language will be introduced. To integrate the creation of a corpus with the process of experimentation and creation of the models that we aim to build in order to address some of the sub-steps of the argumentation mining process, the *ArgMine Framework* was created and a detailed description of the components will be presented.

In sum, we have contributed in the following respects:

- computational models addressing the first two research questions mentioned above;
- critical analysis of the obtained results from the previously mentioned models, to answer the third research question;
- creation of an annotated corpus with arguments from text written in Portuguese;
- an alignment of tools and processes that facilitate and partially automate argumentation mining research (*ArgMine Framework*).

Furthermore, to the best of our knowledge, this is the first work in argumentation mining based on the Portuguese language and, this is the first attempt to build a corpora annotated with arguments obtained from text written in Portuguese.

## 1.3 Thesis Structure

The thesis is structured as follows:

Chapter 2 follows this introduction and presents the state-of-the-art in argumentation mining. It introduces some important concepts and definitions that are central to understand the contents of this thesis. In addition, some of the most influential work is presented.

Chapter 3 describes the ArgMine Framework, one of the major contributions of this work.

Chapter 4 describes the models that were built to address the first two subtasks of the argumentation mining process. To address the first subtask of the argumentation mining process, *Argumentative Sentence Detection* (Section 4.1), a binary classifier is trained to detect argumentative sentences from free text, using semi-supervised machine learning algorithms. From the critical analysis of the obtained results from the task *Argumentative Sentence Detection*, we concluded that the ambiguity associated to lexical clues transformed this intuitive set of features into an irrelevant set of features for the classifiers. In Section 4.2, we investigate if there is an additional property (besides lexical information) that should be verified to consider a word as argumentative keyword (namely, the syntactic role of the word in a given sentence). To address the second subtask of the argumentation mining process, *ADU Boundary Detection* (Section 4.3), a sequential classifier is trained using supervised machine learning algorithms to identify the exact boundaries of the ADU's, given an argumentative sentence.

Finally, Chapter 5 presents the conclusions and points to directions of future work.

# Chapter 2

# State-of-the-Art in Argumentation Mining

This chapter focuses on the concepts and topics that are considered essential to the understanding of the problem, research goals and position of this thesis in relation to other work in the field.

The chapter starts with an introduction to the fundamental concepts about argumentation. Next, some of the most influential argumentation theories for the task of argumentation mining are presented. Afterwards, the detailed argumentation mining process is explained. Then, the state-of-the art on machine learning techniques used to address the task of argumentation mining is presented. Finally, the state-of-the-art on argumentation tools is presented.

## 2.1 Argumentation Fundamentals

In this section some of the fundamental concepts about argumentation are presented. These concepts will be repeatedly mentioned in this thesis and, therefore, a proper definition must be presented.

Argumentation is an ancient and vast topic that has been influenced by many fields such as logic, philosophy and linguistics. Consequently, in the literature, there are several definitions for each of the concepts presented in this section. The following definitions are based on the formulation presented in [MP11, PM09]:

**Definition 2.1. Argumentation** *Argumentation is the process by which arguments are constructed and handled. Handling arguments may involve comparing arguments, evaluating them in some respects, and judging a constellation of arguments and counterarguments to consider whether any of them are warranted according to some principled criterion.*

**Definition 2.2. Proposition** *A Proposition is a simple declarative sentence that has a truth-value (which is either true or false). A proposition is used to make a statement or assertion.*

**Definition 2.3. Argument** *An argument is composed by a set of at least two propositions, being all of them premises, except maximum one, which is a conclusion. The conclusion corresponds to the claim of the argument and can be obtained by one or more reasoning steps (i.e. steps of deduction) from the premises. The premises, also called assumptions, are pieces of evidence offered in support of a conclusion.*

**Definition 2.4. Premise** *A premise is a proposition that is a reason for, or objection against, some claim. A premise is a statement presumed true within the context of an argument toward a conclusion.*

**Definition 2.5. Conclusion** *A conclusion is a proposition that is supposed to be supported by the premises. In the context of ordinary argumentation, the rational acceptability of a disputed conclusion depends on both the truth of the premises and the soundness of the reasoning from the premises to the conclusion.*

In a general form, arguments are justifiable positions that are composed by at least two propositions (the conclusion and, at least one premise). The relation of support between the premises and conclusion is the key characteristic that distinct arguments from other discourse structures.

As an example of an argument, consider the following two sentences:

*"All men are mortal and Socrates is a man. Therefore, Socrates is mortal."*

The conclusion presented in this example is *"Socrates is mortal"* and, the premises are *"All men are mortal"* and *"Socrates is a man"*. In this simple example, the relations of support between the premises and conclusion are evident.

We can graphically represent an argument using an *Argument Diagram*. In an argument diagram, nodes contain propositions (premise or conclusion) and, the indication of support or conflict relations is made using arrows, which connect the premise node to the conclusion node. This representation leads to the visualization of an argument as a graph structure. The corresponding argument diagram for the example previously presented is shown in Figure 2.1.

An argument can be good or bad based on:

- how well the premises support the conclusion;
- the truth-value of the premises.

It is important to realize the difference between the structure of reasoning and the evaluation of the quality of reasoning. In the former, the main concern is related with the principles of building good argument structures and will be the subject of study in this thesis. But these principles do

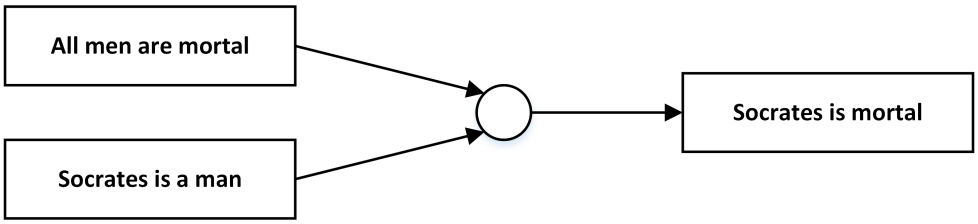


Figure 2.1: Example of an argument diagram

not guarantee that the argument is good. The evaluation of an argument's quality brings further considerations and different research questions that are outside the scope of this thesis.

## 2.2 Argumentation Theory

The discipline of argumentation has ancient roots in dialectics and philosophy and has been influenced by many and diverse areas of knowledge, such as logic, rhetoric, law and computer science. As a consequence, literature in argument representational models is rich and diverse.

The current state-of-the-art in argumentation does not afford a universally accepted theory and, neither a theory that could be applied to every scenario. Currently, there exists a variety of approaches that differ considerably in conceptualization, scope and degree of theoretical refinement [vE01].

In this section, we will introduce some of the most influential proposed annotation theories for argumentation, focusing on theories that highly influenced the development of argumentation mining. Therefore, some important and influential research in argumentation theories that have been designed to fulfill the requirements of other research areas, such as multi-agent systems research, will not be described in this section. For instance, Dung [Dun95] proposed argumentation graphs, which are superficially similar to the kinds of graphs used in this thesis, but were formulated for different purposes. Dung representation of arguments allows for formally modeling the reasoning process but, our concern, on the other hand, is related to the explicit representation of arguments as they are presented in text documents. Similarly, van Eemeren and Grootendorst [vEG04] presented interesting aspects for argumentation but their goals are somewhat different.

One of the first examples to illustrate argumentative processes using argument diagrams was presented by Richard Whately in 1836 [Wha36]. The method described by Whately consists of figuring out the conclusion in the first place, and then trace the reasoning backward, in order to retrieve the grounds (premises) in which the assertion was made. This process can be repeated recursively, obtaining what Whately described as "chain of arguments". The diagram has many of the basic characteristics of modern argument diagrams: statements are represented as nodes and are connected by lines to make up a graph structure [RWM07].

One of the most influential works in the history of argumentation theory was the work developed by Stephen E. Toulmin [Tou58]. Since Toulmin did not agree with the simplistic view of an argument as composed of premises and conclusions, he investigated the actual use of arguments

in order to identify different roles that argument components can play in argumentation.

Toulmin proposed an argumentation scheme containing six functional roles:

- Claim: The statement being argued;
- Data: The facts or evidence used to prove the claim;
- Warrant: The general (and hypothetical) logical statement authorizing our movement from the data to the claim;
- Backing: Statements that serve to support the warrants;
- Rebuttal: Counter-argument or statement recognizing the circumstances when the general argument does not hold true;
- Qualifier: Statements expressing the speaker's degree of force or certainty concerning the claim.

In 1991, Freeman integrated Toulmin's ideas into the argument diagramming techniques of the informal logic tradition [Fre91]. One of the most innovative features introduced by Freeman diagrams was the indication of supposition. A premise can be considered valid only provisionally in order to allow the dialog to continue, and the conclusion can be considered only hypothetical (depending on the stated assumptions). However, one of the most important features introduced by Freeman, in the perspective of argumentation mining, is the distinction between linked and convergent arguments. He recognized two different structures for arguments that should be distinguished in the argument diagram representation:

- convergent arguments: the premises are independent lines of reasoning supporting the conclusion;
- linked arguments: the premises must be combined (they work together) in order to support the conclusion.

In Figure 2.2, the difference in terms of graphical representation between these two kind of argument structures is shown, as defined by Freeman.

Finally, the argumentation theory proposed by Walton defines the theory of argumentation schemes [Wal96]. This theory defines stereotypical patterns of reasoning and has been used extensively for the analysis of argumentative text.

Argumentation schemes are argument forms that represent general inferential structures of arguments. They could be seen as templates for different types of arguments. Arguments found in texts are understood as instances of abstract argumentation schemes. A large catalog of these schemes is provided in [Wal96].

Each argumentation scheme is typically defined by a set of abstract templates for sequence of premises, an abstract template for the conclusion proposition, a set of keywords and a set of


Figure 2.2: Freeman's approach to convergent (left) and linked (right) arguments, [Fre91]

critical questions. Each abstract template of propositions is described in the general case containing some variables that need to be instantiated in a concrete example. One of the most interesting characteristics of argumentation schemes is the set of critical questions associated to each of them. These critical questions can be used to validate the applicability of that specific argumentation scheme to a specific example or, to evaluate a given argument in a particular case and in relation to the context in which the argument occurred. Besides helping in the evaluation of arguments, argumentation schemes can also be used to add missing parts of arguments (often, in the form of implicit propositions, also known as enthymemes). Comparing the argumentation scheme with the argument presented in the text, the missing parts can be easily derived. This observation can have applications in refinement of argument structures or in the detection of missing steps in the reasoning that originated the argument.

From all theories of argumentation that were analyzed, we conclude that most of them assume that the elementary units of an argument can be classified as premises and conclusions. However, over the years, more complex representations have been presented. A detailed overview of the use of argument diagramming techniques to represent the structure of arguments is presented in [RWM07].

So far, theories of argumentation and diagramming techniques were presented to represent the structure of arguments in a general and abstract way. However, when Argumentation Mining is applied to free text, it is dependent of the characteristics and discourse structure of the text. Therefore, further considerations should be made, when it comes to represent the arguments contained in free text. While argumentation theories often assume that the argumentative components are given, a practical and segment-based argumentation mining system has to cope with the linguistic style of the author and the peculiarities of the segmentation process [PS13]. In linguistics, theories of discourse structure have been studied for a long time and several theories have been formalized [Coh87, MT88, Mar00]. Most of these discourse theories assume that any text can be partitioned into a sequence of non-overlapping elementary textual units and that a discourse structure can be associated with the text to represent the relations between elementary textual units. These theories differ mainly in the definition of elementary units and, in the nature and number of different types of relations.

One of the most accepted discourse theories is *Rhetorical Structure Theory (RST)* [MT88].

The authors of this theory formulated a set of 23 rhetorical relations that hold between two elementary units of text, such as evidence, contrast, elaboration, amongst others. Each relation defines a specific role that an elementary unit plays in relation to the other. One of the essential steps in characterizing the discourse structure of a text is to determine the elementary discourse units (EDUs), which are the building blocks of a discourse tree. In RST, the elementary units are classified between *nucleus* and *satellite*. The nucleus is more central to the writer's purpose and is interpretable independently. The satellite is less central and generally is only interpretable with respect to the nucleus. A manual for EDU segmentation that follows the principles of RST is presented in [CM01]. As defined in this annotation manual, the elementary discourse units are non-overlapping spans of text (denominated as "clauses" in the manual). A sentence can contain several EDUs but, on the other side, an EDU cannot contain a span of text from more than one sentence. A set of lexical and syntactic discourse segmentation rules were defined, which are mainly based on discourse markers, conjunctions, verbal forms and punctuation marks. These rules help the annotator in the process of finding the boundaries that separate two adjacent EDUs and, help on understanding the characteristics that are required to consider a span of text as an EDU. For instance, one of the rules indicates that an EDU always has to include a verb. Another rule indicates that if a sentence includes a discourse marker (e.g. "because", "if", "but", amongst others) it has to be separated into two EDUs (if the resulting EDUs include a verb).

The elementary units are related pair-wise and can be hierarchically organized into an entire discourse tree that represents the full text.

Even though the tasks of explaining the coherence of text (the goal of RST) and capturing the argumentative content found in a text are not identical, some researchers employed RST to represent the argumentation structure contained in the text [Gre10, Aza99]. Namely in [Aza99], the author adopted the RST framework, determining the five RST relations (from the twenty-three originally defined by Mann and Thompson [MT88]) that are of interest to represent the argumentative structure. As pointed in [PS13] there are some limitations when adapting the RST to represent the arguments contained in the text.

The formalism followed in this thesis to define the internal structure of an argument determines that each elementary unit can be classified between premise and conclusion. To represent relations between elementary units we define that they can be related using convergent and linked arguments, as introduced by [Fre91].

We have chosen this formalism as the basis of our work due to the following reasons:

- it is the simplest argumentation structure that allows to express the arguments contained within a piece of natural language text;
- it is the most used and accepted argumentation theory by current research in argumentation mining;
- in order to make the process of annotating arguments from text more intuitive for annotators.

## 2.3 Argumentation Mining Process

The full task of argumentation mining can be decomposed into several subtasks. The formulation of the argumentation mining process used in this thesis is based on the work presented in [PS15]. The subtasks to be addressed are the following:

- *Text segmentation:* splitting the original text in elementary discourse units (EDUs). This is a step commonly performed in discourse parsing tasks. Typically an EDU corresponds to a sentence or clause, depending on the task being addressed;
- *Identification of argumentative discourse units (ADUs):* Identification and extraction of the minimal units of arguments (ADUs). It involves discarding argumentatively irrelevant EDUs, joining adjacent EDUs to form larger ADUs or partition of EDUs into several ADUs;
- *ADU type classification:* determining the type of argumentative unit. Classification of each ADU into the different types of argument components;
- *Relation identification:* establish the relations between individual ADUs, leading to a set of incomplete argument diagrams (the arrows are unlabeled for each argument diagram);
- *Relation type classification:* classify the type of argumentative relation, leading to a set of complete argument diagrams.

In the *Text segmentation* subtask, typically performed in natural language problems, is similar in nature to that of finding the elementary discourse units in discourse parsing. Commonly, sentence boundaries are considered to be EDU boundaries but, in addition, complex sentences may be broken into several EDUs, which generally correspond to clauses.

For the specific case of argumentation mining, [PS13] refers to these “minimal unit of analysis” as Argumentative Discourse Units (ADUs).

**Definition 2.6 Argumentative Discourse Units (ADUs)** *Argumentative Discourse Units are non-overlapping spans of text corresponding to the minimal units of arguments.*

An ADU may not always be as small as an EDU, for example, "when two EDUs are joined by some coherence relation that is irrelevant for argumentation, the resulting complex might be the better ADU, when it collectively plays some specific role in the argumentation." [PS13, p. 21]. However, on the other side, it can also happen that an EDU should be broken in several ADUs, for example, when premises and conclusions are presented in the same sentence. For this reason, methods to automatically identify and extract ADUs from free text or methods that extract the ADUs from the EDUs previously obtained, have both to be explored specifically for the task of argumentation mining (*Identification of argumentative discourse units* subtask).

Following the formalism to represent an argument that will be used in this thesis, ADUs can be further classified between premise or conclusion, depending on the role that each ADU is playing

in the argument (*ADU type classification* subtask). Therefore, the definition of ADU used in this thesis is completed with the definitions of *Premise* (Definition 2.4) and *Conclusion* (Definition 2.5) presented in Section 2.1.

The last two substeps of the argumentation mining process are related to the connection between premises and conclusions in order to build the final argument diagram. First, from all the premises and conclusions extracted from the text in the previous substeps, we have to connect premises to the conclusions that they support or attack (*Relation Identification* subtask). After this step, we will obtain a set of incomplete argument structures. Finally, in the subtask *Relation type classification*, each connection is classified between the type of relations defined by the formalism used to represent the argument structure. Following the argumentation structure that will be used in this thesis, we aim to classify each relation between support or attack. A relation of support means that the premise contains evidence in support of the conclusion and, on the other side, a relation of attack means that the premise presents evidence that contradict the conclusion.

In this thesis, we will address the first two subtasks of the argumentation mining process described above. Therefore, in the end, instead of the complete argument diagram we aim to identify and extract the argumentative discourse units presented in the text. This implies determining the sentences of the text that contain argumentative content and the identification of the exact boundaries of the argumentative discourse units presented in argumentative text.

## 2.4 Approaches to Argumentation Mining

The aim of machine learning is the study of algorithms that at some task $T$ improve their performance $P$, based on the experience $E$ [Mit97]. Based on a set of examples, a learning algorithm creates a model relating outputs to inputs that fits these examples and, the goal is to learn a model that generalizes well for unseen examples.

Next, we will introduce the current state-of-the-art in relation to the approaches that have been used to address the task of argumentation mining. Current approaches to automatic analysis of argumentative content in text usually follow the supervised machine learning paradigm. After, we present the current state-of-the-art using semi-supervised machine learning techniques and we motivate the use of this paradigm to approach the problems that we are trying to solve.

### 2.4.1 Supervised Learning

The aim of supervised machine learning is to create a model that maps inputs to outputs given a set of input and desired output pairs. The supervised machine learning algorithm analyzes the training examples and infers a function, which can be used to map new examples. In order to obtain good performances in unseen data, the model should be able to generalize well from the training data.

One of the first works devoted to the identification of arguments in text was *Argumentative Zoning* [TM02], whose aim is the segmentation of a discourse into discourse segments or zones, each playing a specific rhetoric role in the text. Teufel *et al.* presented an algorithm which, on the

basis of the annotated scientific articles, classifies the content into a fixed set of seven rhetorical categories.

The aim of argumentation mining is different from argumentative zoning. In the former, we are not only interested in classifying each text segment by their argumentative function, but we also aim to automatically identify the argumentative relations between each argumentative component, leading to the detection of full argument diagrams.

One of the first attempts to apply this kind of techniques to the task of argumentation mining and, one the most influential is presented in [MBPR07, PM09]. In their work, they analyze the main research questions of the entire argumentation mining process. Also, they present the different methods studied and developed in order to approach the challenges of argumentation mining applied to legal texts. They used two corpora to develop their work: the Araucaria corpus [Ree06] and the ECHR corpus. The first corpus is a general corpus where the data was collected from 19 newspapers, 4 parliamentary records, 5 court reports, 6 magazines and 14 further online discussion boards and “cause" sources. The second corpus is composed by a set of documents extracted from legal texts of the European Court of Human Rights (ECHR). The ECHR developed a standard type of reasoning and structure of argumentation. Therefore, its documents are an interesting test set for argumentation analysis. The set of annotated arguments from both corpora were manually annotated by experts.

The first task addressed by Mochales and Moens, the detection of argumentative sentences, is seen as a classification problem. In their work, they made the assumption that the *ADUs* are complete sentences. Due to the fact that legal texts “normally present premise and conclusion in subordinate sentences or independent sentences instead of subclauses" [PM09, p. 101], they argue that this assumption is viable. Then, each sentence is represented as a vector of features and a classifier is trained on examples that were manually annotated. They used generic features involving lexical, syntactic, semantic and discourse properties that can easily be extracted from the texts. The detailed list of features can be found in [PM09]. They reported 73% of accuracy in the Araucaria corpus and 80% of accuracy in the ECHR corpus, using the maximum entropy model.

Next, Mochales and Moens studied the classification of each argumentative proposition into premise and conclusion. Again, their approach is to formalize this problem as a binary classification problem and to work with statistical classifiers. For this task, they used a more sophisticated set of features. They preserved three features from the previous task related to the general structure of the text and sentence. In addition, they introduced features related to the argumentative category (among 5 categories defined by the authors of this work), a binary feature indicating the presence of a reference to a law article in the sentence (typically found in premises of legal texts), the type of rhetorical pattern occurring on current, previous and next sentence among 5 types defined by the authors (Support, Against, Conclusion, Other or None), amongst others. The complete description of all the features used can be found in [PM09]. They reported F1-score of $68,12\%$ and $74,07\%$ for proposition classification into premise or conclusion respectively, using a support vector machine (SVM) as classifier.

For the last task, detection of the argumentation structure, Mochales and Moens used a different approach. Instead of a machine learning approach, they manually derived rules that are grouped into a context-free grammar (CFG). Since argumentative parsing is a difficult task, they restricted their research to a limited complexity. They focus on the legal domain and they derived the rules for the CFG based on information extracted from 10 ECHR documents. They reported 60% accuracy in detecting the argumentation structures.

As the field of argumentation mining continued its growth, an increasing number of contributions and different methods have been explored by the community in the last years.

Rosenthal and McKeown [RM12] automatically determined whether a sentence is a claim using logistic regression. They used lexical and sentiment-related features and achieved accuracies between 66% and 71%. Park and Cardie [PC14] classified propositions in user comments into three classes (verifiable experiential, verifiable non-experiential and unverifiable) using SVM and reached 0.69 macro F1-score. Goudas *et al.* [GLPK14] identified premises in Greek social media texts using a two-step approach. In the first step, they classified each sentence as "sentence containing arguments" and "sentence that don't contain arguments" (argumentative sentence detection). In the second step, they try to identify the exact fragments that contain the premises. They represented each argumentative sentence using *BIO* encoding. The *BIO* encoding seeks to classify each token with a single tag from the following set:

- "B-X" represents the begin of a segment of type X. It must be applied on the first token of a segment;
- "I-X" represents a token as being inside a segment of type X. It must be applied on any token inside a segment, except the first and last ones;
- "O" represents a token as being outside a segment. It must be applied on any token that is not contained inside a segment.

Utilizing conditional random fields as sequential model, they achieved 0.42 F1-score for identification of premises. Boltužić and Snajder [Bn15] employed hierarchical clustering to cluster arguments in online debates using embeddings projection, performing intrinsic evaluation of the clusters. Rooney *et al.* [RWB12] classified sentences into four categories (conclusion, premise, conclusion-premise and none) achieving 0.65 of accuracy. They worked with the Araucaria corpus and they assume the text is already segmented into argument components. Stab and Gurevych [SG14] classified argument components into four categories (premise, claim, major claim, non-argumentative) using SVM and, achieved 0.73 macro F1-score. Also, they classified argument relations (support and attack) reaching 0.72 macro F1-score. They worked in a dataset of persuasive essays.

Reed and Lawrence, in [LR15], demonstrated that combining different techniques can lead to significant increases in performance for the task of argumentation mining comparing to the performance of any individual technique. These results contrast with some other areas of text mining and machine learning where combining different techniques is either not possible or else yields

only marginal improvements. First, they explored individually three different techniques that have been applied to this problem and drawn some conclusions from the analysis of their performances. The techniques explored were supervised machine learning using argumentation schemes, topic modeling and purely linguistic methods. From the strengths and weaknesses observed for each technique, they conceived an algorithm that combined each of the techniques previously mentioned. They achieved F1-score of 0.83 for determining connections between propositions and reported results that are very close to a manual analysis of the same text. The results are based on correctly identified connections when compared to the manual analysis. Notice that these results were obtained considering the task of determining the argumentative structure from a piece of text which has already been split into its component propositions.

Then, in the same work [LR15], Reed and Lawrence addressed the subtask of identification of the *ADUs* using a technique called *Propositional Boundary Learning* [LRA+14]. This technique uses two naïve Bayes classifiers, one of them to determine the first word of a proposition and, the other, to determine the last word. First, the original text is split into words. After, a list of features is calculated for each word. The detailed list of features used can be found in [LR15]. Then, the classifiers are trained using a set of manually annotated training data. Using this method they reported a 32% increase in accuracy over simply segmenting the text into sentences, when compared to argumentative components identified by a manual analysis process.

In order to be able to learn, supervised machine learning algorithms need a considerable amount of labeled data. However, annotating arguments in discourse is costly, error-prone and requires some training or basic knowledge of argumentation concepts. Therefore, it is very difficult to get an annotated corpora with a sufficient number of annotations to train supervised machine learning algorithms. This is one of the biggest challenges that the argumentation mining community faces currently. Another problem reported by several researchers is that the models are domain dependent. Since some of the features used in current state-of-the-art implementations are based on lexical information (such as keywords, cue phrases, word couples, etc) and the corpora used to train the algorithms are domain specific, the models are unable to generalize well for texts from other domains. To overcome this problem, semi-supervised machine learning algorithms can be applied to gain more information by exploring unlabeled datasets.

Next, we will present the state-of-the-art using semi-supervised learning algorithms for argumentation mining.

### 2.4.2 Semi-Supervised Learning

Semi-supervised falls between unsupervised learning (without labeled training examples) and supervised learning (with labeled training examples). In this case, some of the training examples are labeled and others are not labeled. Using semi-supervised learning methods we are able to make use of this additional unlabeled data to better generalize the models to unseen data.

Habernal and Gurevych, in [HG15], studied whether leveraging unlabeled data in a semi-supervised manner can boost the performance of argument component identification and to which extent is the approach independent of domain and register. This corresponds to one of the first works attempting to explore the semi-supervised machine learning paradigm in argumentation mining.

The argumentation model used was a variation of the Toulmin model presented in Section 2.2. There are five different components in this model: claim, premise, backing, rebuttal and refutation. Relations between argument components are implicitly encoded in the model. They approach the task of identification of the *ADUs* as a sequence tagging problem and employ $SVM^{hmm}$. They represent the original text using *BIO* encoding. Therefore, each token is labeled with one of the 11 possible labels (5 types of argument components * (B or I tag) + one O tag).

They proposed novel features that exploit clustering of unlabeled data from debate portals based on a word embeddings representation.

They divided the feature set into what they call baseline features and unsupervised features. The baseline features are features typically used in the supervised machine learning methods: Lexical baseline (FS0), structural and syntactic features (FS1), sentiment and topic features (FS2), semantic and discourse features (FS3). They enrich the previously mentioned set of features with "unsupervised features", obtained from external large unlabeled resources. They assume that the posts from unlabeled debate portals contain valuable information that will help classifying arguments in labeled data. In order to do so, they employ clustering based on latent semantics, which they formalize as argument space features (FS4). They take data from the debate portals, project them into a latent space using word embeddings and cluster them. They observed that vectors belonging to the same cluster in the latent vector space exhibit some interesting properties, such as semantic similarity. Then, they project each sentence in the labeled data to the latent vector space, compute its distance vector to all cluster centroids, and encode this distance vector directly as real-valued features. Therefore, each sentence can be labeled as belonging to each cluster with a certain weight.

Using this method they significantly improved the performance on the task of *ADU* type classification and outperform some state-of-the-art baselines. While the performance of the argumentation mining system decreased in cross-validation scenario, they gained almost 100% improvement in cross-domain and cross-register settings (different type of data such as news articles, forum posts, blogs, amongst others).

## 2.5 Argumentation Tools

In this section, some state-of-the-art tools for argumentation will be presented.

*OVA* (Online Visualization of Argument) [1] is a browser based tool developed by *ARG-tech: Center for Argument Technology* [2] that is hosted at the *University of Dundee* that supports the

[1] http://ova.computing.dundee.ac.uk
[2] http://www.arg-tech.org/

visualization and analysis of arguments from text in a web interface and follows the most recent standards on representation of arguments. This tool provides a drag-and-drop interface to analyze arguments from text and visualize the corresponding argument diagram in a user-friendly and intuitive way. We can save the resulting annotation (argument diagram that represents all the arguments contained in the text) in a *JSON* file, where the information is structured according to *AIF* (Argument Interchange Format). Also, we can save it directly into *AIFdb* [3].

*AIF* is a specification to represent the structure of arguments that was the result of an international effort to develop one specification that could allow the exchange of argumentation resources between different platforms and tools [CMM+06].

*AIFdb* is a database that allows the storage and retrieval of *AIF* compliant argument structures. *AIFdb* offers some web services allowing the user to interact with the stored arguments. For instance, it offers a search interface to locate and visualize the arguments contained in the database. In addition, this resource is commonly used by researchers to store the annotations, and to make the annotations publicly available for other researchers in the community.

Other argument diagramming tools exist beside *OVA*, such as *Rationale* [4], *Carneades* [5], *iLogos*[6], amongst others. From our understanding, *OVA* is the most user-friendly and intuitive tool to annotate arguments from text, it allows to save the annotations in a standard format, is web based (the annotators do not have to download any specific software, they can do everything on the web) and, because the argumentation diagramming process is according to the argumentation theory that will be used in this thesis, *OVA* was the argument diagramming tool chosen to integrate with the annotation platform of the ArgMine Framework, which we describe in Chapter 3.

[3] http://www.aifdb.org
[4] http://www.reasoninglab.com/rationale/
[5] http://carneades.github.io/
[6] http://www.phil.cmu.edu/projects/argument_mapping/

# Chapter 3

# ArgMine Framework

This chapter presents the *ArgMine Framework*, which aims to integrate the process of creating an annotated corpus with arguments and the semi-automated process of selection and experimentation of different models and relevant features in different steps of the argumentation mining process.

Typical approaches to argumentation mining follow the supervised machine learning paradigm, in which a set of labeled data (corpus) is necessary in order to build a model that learns how to map inputs to the desired outputs. For the task of argumentation mining, the inputs are natural language texts, and the outputs are argument diagrams, which represent the structure of the arguments contained within the text. In this thesis, we aim to study argumentation mining applied to texts written in the Portuguese language. Therefore, a corpus containing texts annotated with arguments in the Portuguese language is required. However, to the best of our knowledge, no such corpus exists. Thus, we have created an annotation platform, in which the creation of an annotated corpus with arguments from texts written in the Portuguese language is taking place. A detailed description of this annotation platform can be found in Section 3.1.

As described in Section 2.3, the argumentation mining process can be divided in several subtasks. Each subtask is related to a specific problem in the whole argumentation mining process and, the creation of models that can be used to automatically execute each of the subtasks may require the exploration of different machine learning and natural language methods. Besides the differences between each of the subtasks, all of them have to be integrated with the annotation platform, from which the data that will be used to train the machine learning algorithms is generated. Both processes, the creation of an annotated corpus and the creation of models to address some of the subtasks of the argumentation mining process, are in a initial stage and being developed simultaneously, which poses several challenges. One of the main challenges is given by the fact that at the moment that this document was written there was no available corpus containing text documents written in Portuguese and annotated with arguments, which has the following characteristics: stability and inclusion of high quantity and quality of annotations. In the ArgMine corpus, which will be described in Section 3.2, the number of available annotations is low, al-

though increasing over time. Moreover, the quality of the available data may also be changing over time. Since the number of available annotations is low, it is necessary to continually update the dataset used by machine learning algorithms in order to increase the number of examples from which they can learn in a manner that is useful to address a specific task. In this dynamic scenario, some of the assumptions made when creating the models or even what the models learned from the previous version of the dataset, can be not valid for an updated version of the dataset. For this reason, different methods may be required and new experimentations need to be run. In this way, the semi-automation of the process of experimentation can be helpful in the development of models for each of the subtasks in the argumentation mining process.

For these reasons, the ArgMine Framework was created. This framework aims to integrate the creation of a corpus containing news articles annotated with arguments in the Portuguese language and the semi-automated process of selection and experimentation of different models and relevant features for different steps in the argumentation mining process [RCT16].

The main guideline followed for the implementation of the ArgMine Framework was that it should be modular and easy to adapt for different tasks in the argumentation mining process. It should be modular in order to allow experimentation of some components of the framework maintaining the remaining components unchanged. This is particularly important when dealing with a changing corpus and when we aim to understand the influence of some components in the system. In future work, we expect to integrate all the models obtained after addressing each subtask of the argumentation mining process, to automatically identify and extract the arguments contained in texts written in the Portuguese language. In this sense, we aim to continually use this framework when addressing each of the subtasks mentioned in Section 2.3.

The main programming language chosen for the implementation of the framework was *Python* due to the following reasons:

- this programming language has been widely used by the natural language processing community and, therefore, some useful linguistic resources are publicly available;
- useful tools to work with machine learning algorithms and techniques are available, such as, *scikit-learn* [PVG+11], *NumPy*, *SciPy*, *matplotlib*, *CRFsuite*, amongst others;
- Python is relatively easy to work with text documents, *HTML*, *XML* and *JSON* files, which are useful to process some of the resources that have to be used in this framework.

To implement the annotation platform, the following programming languages were also used: *SQLite*, *HTML*, *CSS*, *PHP* and *JavaScript*.

In terms of architectural pattern, the framework follows the *Pipes and Filters* pattern, in which a sequence of processing components (Pipes), each performing a specific function, are connected by channels (Filters). Each component receives the data in a specific format from the previous component in the pipeline, transforms it according to its specific function and outputs the resulting data to be processed by the next component in the pipeline.

Designing the framework following the Pipes and Filters architectural pattern has the following advantages:

- captures the natural process of a software implementation that performs natural language processing and uses machine learning algorithms. The process is typically characterized by a set of transformations that are made to the original natural language text in order to transform it into a set of representative features suitable as input for machine learning algorithms and techniques;
- allows the reuse of some components that are common to different steps in the argumentation mining process, facilitating the integration and adaptation between different subtasks of the argumentation mining process. For instance, the integration of the *ArgMine corpus* with the *Data Preparation* component in the *Machine Learning Module*;
- allows the easy integration and experimentation of different methods in each of the components of the framework, without requiring changes in other components.

The *ArgMine Framework* is composed by a set of components depicted in Figure 3.1. We can divide the complete set of components in two high-level modules:

- *Corpus Creation Module*: set of components related to the creation of an annotated corpus with arguments from Portuguese news articles;
- *Machine Learning Module*: set of components related to the natural language processing and machine learning process.

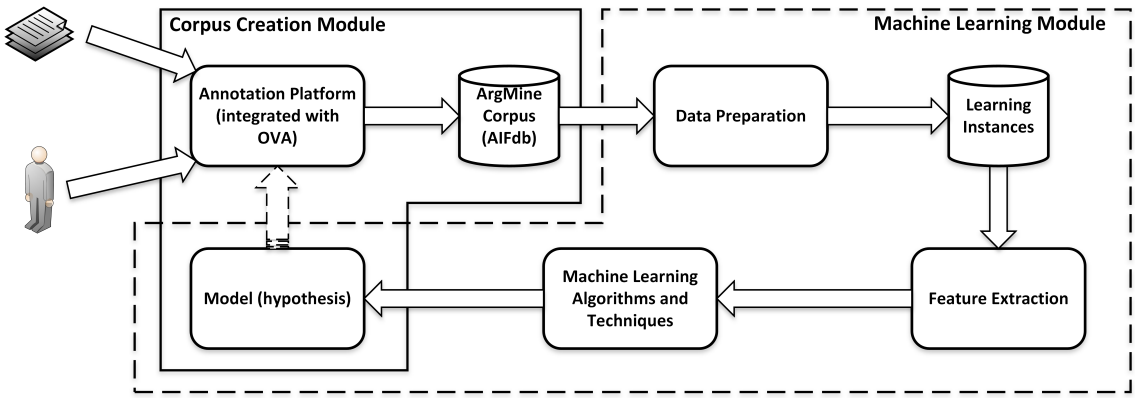


Figure 3.1: ArgMine Framework general architecture. The full-line area delimits the Corpus Creation Module, while the dashed-line area delimits the Machine Learning Module.

In the following sections, each component of the ArgMine Framework is described. Section 3.1 is dedicated to the description of the *Annotation Platform component* and Section 3.2 to the description of the *ArgMine Corpus*, both from the *Corpus Creation Module*. In Section 3.3 a description of the *Machine Learning Module* is presented. Finally, Section 3.4 describes the integration of the models created to address some of the subtasks of the argumentation mining process in the *Annotation Platform*.

## 3.1 Annotation Platform

Supervised machine learning algorithms need a set of labeled data (corpus) in order to build a model that learns how to map inputs to the desired outputs. For the task of argumentation mining, the inputs are natural language texts written in Portuguese and the desired outputs are argument diagrams corresponding to the structure of the arguments contained within the text.

To the best of our knowledge, an available corpora satisfying the requirements previously described does not exist. Since these annotations are required in order to apply supervised machine learning algorithms, we started to gather annotations from human annotators. The outputs of these annotations are argument diagrams, which are graph structures. In order to facilitate the process of annotation an intuitive tool to represent such argument diagrams is required. Thus, we have created an online and publicly available platform[1] where annotators can access news articles written in Portuguese and, in an intuitive way, identify and annotate the arguments presented in the text in the form of argument diagrams. In this annotation platform, it is also possible to find educational material that was built with the purpose of presenting valuable information to the annotators about the topic, explaining how to use the platform and to give some tips about argument diagramming, as shown in Figure 3.2.


Figure 3.2: “Anotar Argumentos" section of the Annotation Platform, in which some information related to the annotation process is available.

In order to build this platform, we performed a study of the current state-of-the-art on argumentation tools that could be useful for this task, as described in Section 2.5.

From all the tools analyzed, we decided to integrate the annotation platform with two external

[1] https://web.fe.up.pt/~ei11124/argmine/

tools, namely *OVA* [JLR14] and *AIFdb* [LR14]. The main advantages to use these tools are the following:

- the annotation process can be performed completely online (which avoids local installations and software setups that could difficult the annotation process);
- *OVA* interface allows an intuitive annotation of arguments from text (using a drag-and-drop interface);
- *OVA* offers the possibility to save the resulting annotation in the *AIF* standard and to save these annotations in the *AIFdb*;
- *AIFdb* offers web services that allows us to search and visualize the collection of annotations;
- both tools have been widely used and are well recognized in the argumentation mining community.

In the annotation platform, each annotator can find a collection of news articles, namely opinion articles. The news articles were provided by *SAPO Labs*[2]. From the different types and sources of news provided by *SAPO Labs* we focused on opinion articles. An opinion article is an article, published in a newspaper, that reflects the author's opinion about a specific subject. One of the advantages to work with this type of news articles, in relation to other types of articles, is the richer argumentative content that is typically presented in an opinion article text. On the other hand, one of the disadvantages is the fact that authors tend to use refined vocabulary which can make the interpretation of the text more challenging. In addition, different authors tend to use different writing styles which creates some variability in the analyzed texts, and in turn complicates the task to the machine learning algorithms. Another typical characteristic of opinion articles is the length of the articles, which are typically longer comparing to other types of news articles. Since the length of text can increase the time consumed in the annotation of each opinion article, we decided to present the opinion articles segmented by paragraphs in order to reduce the complexity of the task. This means that each article is divided in several parts (by paragraph) and each part is presented as a different annotation document to the annotator. This decision can have some drawbacks: when the argument is spread in several paragraphs, it is impossible to annotate it because each part of the argument will be presented in different documents; moreover, in some situations it can happen that some information in the remaining parts of the document could be useful and/or necessary to detect the arguments presented in one of the parts of the document. In the first case, we assume that this situation will not occur too often. A paragraph corresponds to a distinct section in a document, usually dealing with a single theme and terminated by a new line. Since arguments have to be about some topic and changes in topic can indicate that different arguments are being expressed, as explored in [LR15], then this assumption seems reasonable. However, in some situations, arguments can in fact be spread in several paragraphs. Typically, in these situations, the

[2] http://labs.sapo.pt/

arguments that are being presented require complex reasoning and knowledge about the world, since they lack of lexical and syntactic cues (because the parts of each argument are not presented in a contiguous sequence). This kind of arguments are very difficult to address. Since, in this thesis, we assume that argumentative discourse units occur in a contiguous sequence, this kind of arguments are out of scope for the models that we aim to build at this moment. In the second case, negative consequences of this decision were already experienced during the development of this thesis. In these situations some information contained in other paragraphs of the news article are essential or, at least, helpful to the understanding of the arguments in a specific paragraph. For instance, pronouns that are presented in the current paragraph which mention some entity referred in other paragraph in the same news article. Not knowing the entity that the pronoun is referring to, can make the detection and identification of the arguments contained in the paragraph more difficult and/or ambiguous.

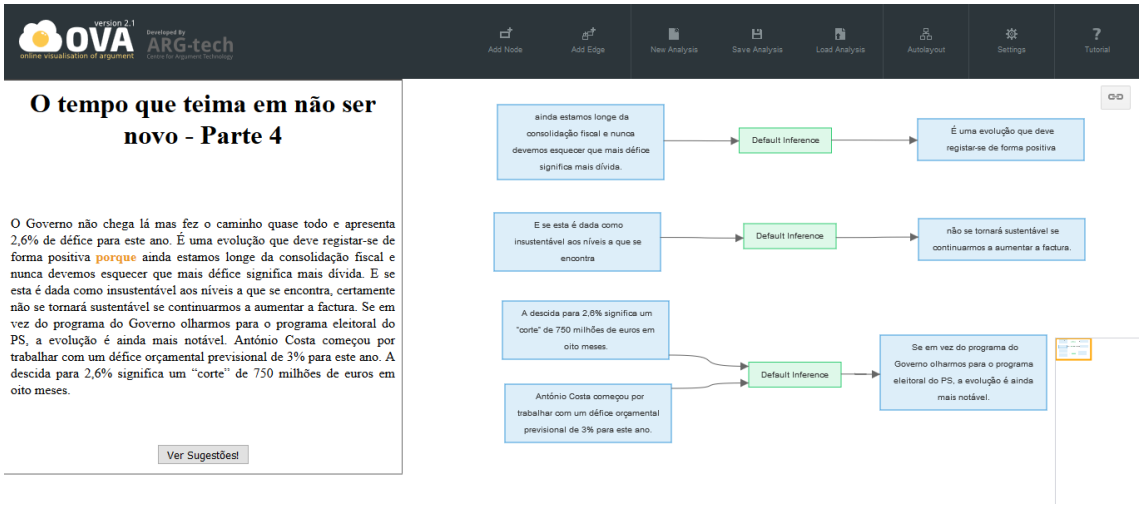


Figure 3.3: Opinion article annotated with arguments

In the annotation platform, after selecting a particular news article, the annotator is redirected to the *OVA* interface, as shown in Figure 3.3. In the left side of the *OVA* interface, the selected news article is presented. As explained in Section 2.5, the drag-and-drop interface of *OVA* allows users to select a piece of text presented in the left side of the interface (drag) and, after clicking the interface in the right side, the selected text will be inserted in a rectangular figure (representing a node in the argument diagram) on the right side of the interface (drop). In order to complete the annotation, the interface allows the annotator to connect nodes with arrows, which can be used to indicate relations of support or conflict between two nodes. When the annotation of the arguments contained within the text is completed, the annotator can save the corresponding argument diagram in *AIFdb*. We created a corpus in the *AIFdb* database, the *ArgMine Corpus*[3], in which all the annotations obtained from the annotation platform are being collected. Storing the annotations

[3] http://www.arg.dundee.ac.uk/aif-corpora/ArgMine

in this database have the following advantages: it allows us to save and retrieve the annotations in the standard *Argument Interchange Format (AIF)* and it allows the online and free access to this corpus by researchers in the community.

## 3.2 ArgMine Corpus

In order to successfully apply machine learning algorithms, a good training corpus is crucial. The training corpus contains the instances (or examples) from which machine learning algorithms generate models to automatically address some specific task. Therefore, creating a collection of annotated documents (corpus) with arguments is an important requirement in order to address any subtask in the argumentation mining process.

Constructing an annotated corpora is, in general, a complex and time-consuming task, which can require knowledge of human experts in the field and a platform to facilitate the annotation process. Providing a set of guidelines that should be followed by annotators is a means to ensure that this task is performed with a shared understanding of the problem and expected outputs by every annotator. Creating a corpus for argumentation mining is very challenging, due to the following characteristics of the task [LT16, HEKG14, MI09]:

- the identification and detection of arguments from text is complex and requires good interpretation skills;

- constructing argument diagrams from text, which requires understanding the arguments that are being made and how each of the components of the argument are related to each other, can be quite complicated (and controversial in some situations) even for humans;

- the fuzziness of unit boundaries can hamper some stages of the argumentation mining process (e.g. transforming annotation into training instances for a specific task) and inter-annotator agreement (IAA) metrics.

Addressing a complex task, such as argumentation mining, using machine learning algorithms requires a corpus with high quality and quantity of annotations. To measure the quality of the obtained annotations, inter-annotator agreement measures are typically employed. These measures require that different annotators should annotate the same document, in order to evaluate the agreement shared by different annotators in relation to the annotation of a particular document.

In order to create an argumentation corpus, there are some characteristics of the corpus that have to be defined according to the characteristics of the tasks that require the annotated corpus [LT16]: granularity of input, genre of input, argument model, granularity of target, and goal of analysis. The *ArgMine Corpus* was created targeting the complete argumentation mining process and can be described with the following characteristics:

- genre of input: news written in Portuguese, including general news articles (without any filter) and opinion articles which are presented by paragraph. Topics and writing styles may vary between different articles;
- granularity of input: we consider the detail of analysis at the intra-sentence level in order to allow annotations of intra-sentence argumentative discourse units boundaries;
- argument model: to represent the internal structure of an argument we follow the conclusion/premise model and, relations between ADUs are represented using convergent and linked arguments, as defined in Section 2.2;
- target: full argument diagram, in which premises and conclusions are connected using convergent or linked arguments;
- goal: argument detection, ADU boundaries identification, ADU classification and relation prediction.

The *ArgMine Corpus* is at an early creation stage. At the time of this writing, the *ArgMine Corpus* contains a total of 215 annotations: each annotation corresponds to the complete argument diagram of all the arguments presented in a text document (e.g. paragraph of an opinion article).

Due to the low number of annotations that were submitted to the *ArgMine Corpus* and due to the characteristics of annotating arguments from text that were previously described (such as, fuzziness of argument components boundaries, controversial task, complex representation of the arguments contained in a text document), we did not calculate inter-annotator agreement metrics for the current state of the *ArgMine Corpus*. Instead, a manual analysis of the obtained annotations was made in order to evaluate their quality. From this analysis, we can conclude that the annotations that will be considered, in this thesis, to address some steps in the argumentation mining process have good quality but low inter-annotator agreement, since the annotators often agree in the main (or more evident) arguments that are being made in the text but, on the other side, often disagree in some of the components or arguments that are more difficult to identify or, that require a fine-grained analysis of the arguments that are being made. We also observed that in some situations even if annotators tended to identify essentially the same argumentative component, the boundaries differed slightly. For instance, one of the most common situations occurs when exists a connector separating two argumentative components. Some annotators tend to include the connector in the argumentative component and, on the other hand, some annotators do not include the connector in the argumentative component.

From the analysis of the current state of the *ArgMine Corpus* and from the annotation process that was made, the following lessons were learned:

- working with news articles, which can have different topics, is a challenging task and often requires common sense and knowledge about the topic. In this kind of text documents, writers tend to omit some clue words and well known facts in order to avoid text repetition and to capture the readers attention. Instead, they often appeal to the common sense and knowledge about the world that they assume that the human readers have;

- opinion articles contain more argumentative content than other kinds of news articles;
- opinion articles are difficult to interpret comparing with other types of news articles and, the vocabulary that is used is more refined and varied. This can be problematic in terms of learning because it makes the task of retrieving patterns in data more challenging.

## 3.3 Machine Learning Module

This section presents the *Machine Learning Module* of the *ArgMine Framework*, which aims to semi-automate the process of selection and experimentation of different models and relevant features in different steps of the argumentation mining process.

This module is composed of a set of components, which encapsulates some of the typical methods that have to be applied when solving tasks that require the use of natural language processing techniques and machine learning algorithms. In this section, a general description of this module is presented. Since different steps in the argumentation mining process may require different tools, techniques and algorithms, this module should be instantiated for each of the steps that are addressed in the argumentation mining process. Moreover, even for a specific stage of the process, experimenting different configurations of each component is also desired. Consequently, a modular architecture of the entire process which facilitates the process of experimentation is one of the desired properties for this module. This is particularly relevant if this process is performed using a dataset containing a small number of instances and that is continually growing. In this situations, the characteristics of the corpus may also be changing and, therefore, techniques and/or algorithms that were good to model one version of the dataset may not be good to model another bigger version of the dataset. In such an evolving setting, the process of selection and experimentation of models and features is time-consuming and repetitive. The aim of the Machine Learning module is to semi-automate this process and to facilitate the maintainability of code when experimenting different configurations of the machine learning algorithms and techniques.

The Machine Learning module is composed of three components (see Figure 3.1): *Data Preparation*, *Feature Extraction* and *Machine Learning Algorithms and Techniques*.

In the *Data Preparation* component, a collection of documents annotated with arguments (*ArgMine Corpus*) is received as input. Each instance of the *ArgMine Corpus* is composed by two documents: a text document containing the complete news article and the *AIF* file containing the corresponding argument diagram, obtained from the annotation platform (as described in Section 3.1). In this component, each instance of the *ArgMine Corpus* is analyzed and transformed into a training instance, which will be the input of the learning process, according to the task that is being addressed. This may involve the following steps:

- filtering irrelevant annotations for the task at hand (e.g. annotations that do not contain certain components which are the target of the current task);
- applying natural language processing techniques to process the input data (e.g. tokenization);

- transforming the original text and annotation into a preprocessed text which could be more suited for the task at hand; and

- combining the annotation (argument diagram) with the original text document to transform the annotation into a (set of) training instance(s) (depending on the task at hand), which may offer several difficulties due to the fuzziness of the boundaries in the annotated text.

After all the transformations made in this component, the resulting dataset contains a set of training instances that will be used by the remaining components in this module in order to proceed with the learning process for the specific task that is being addressed.

The *Feature Extraction* component is used to transform a set of training instances (data received as input) into a set of numerical features that best represents the data, in a format supported by machine learning algorithms. For each subtask of the argumentation mining process a set of different features extraction methods may be required in order to better represent the dataset, resulting in a set of features that must be combined together to create the final feature set. Each feature set must be normalized or scaled, according to the values that it generates. This preprocessing step (normalize or scale the feature set) is a common requirement for many machine learning algorithms and, avoids that different ranges of values that features might have, influence in the importance that the classifier will assign to each feature. In terms of implementation, the union of different feature sets is made using the *FeatureUnion* class provided in *scikit-learn*, which allows to concatenate the result of several transformer objects into a single transformer. Each feature extraction method must be implemented using the conventions defined in the *Transformer API*, as described in the *scikit-learn* documentation.

In the *Machine Learning Algorithms and Techniques* component, a set of feature analysis techniques and estimators are applied to the feature set provided by the previous component in the framework with the aim to determine the subset of variables and the model (and corresponding parameters) that performs better in a specific subtask of the argumentation mining process. Feature analysis techniques are commonly used to perform dimensionality reduction by choosing the subset of features that best explain the data (feature selection methods) or, by creating new features (latent variables) that are combinations of existing and correlated features (feature extraction methods). Estimators correspond to machine learning algorithms that given a set of training instances (training set) learn how to create a model that maps inputs to the desired outputs.

These last two components, *Feature Extraction* component and *Machine Learning Algorithms and Techniques* component, are combined in a Pipeline object (from scikit-learn), which applies a set of transforms to the data received as input and then fits the final estimator to the obtained feature set. Combining these two components in a single Pipeline object allows us to assemble several steps (transformation and final prediction) that can be cross-validated together while setting different parameters.

Finally, parameters that are not directly learnt by estimators can be automatically set by searching a parameter space, in order to find the combination of configurations that achieves the best performance score. These parameters that are not directly optimized during the learning procedure,

but contain some variables that have to be defined, are commonly known as *hyperparameters*. Typical examples of hyperparameters are the parameter *C*, *kernel* and *gamma* to setup a *Support Vector Machine*, the number of components in *Principal Component Analysis*, and so on. After definition of the target parameter space, the hyperparameter optimization is made using methods provided in *scikit-learn* (namely, *RandomizedSearchCV*).

In the end, the final model is obtained, which can be applied to make predictions in unannotated texts for the subtask in the argumentation mining process that is being addressed.

## 3.4 Models Integration

The aim of argumentation mining is the automatic identification of the argumentative structure contained within a piece of natural language text. As described in Section 2.3, the entire process can be divided in several subtasks, each of them describing some of the sub-problems that have to be formulated in order to successfully address such a complex task as argumentation mining.

As shown in Figure 3.1, and represented with a dashed arrow between the *Annotation Platform* component and the *Model (Hypothesis)* component, we aim to display the resulting predictive capabilities of the models that are developed to address some of the sub-steps of the argumentation mining process directly in the *ArgMine Platform*. Displaying the obtained predictions directly in the *Annotation Platform* allows us to demonstrate the predictive capabilities of the models developed during the process and, in a intuitive way, to visualize the obtained predictions. In addition, the predictive capabilities of the developed models can be used to help the annotator during the annotation process. If the user clicks in the button named *"Ver Sugestões!"* (see Figure 3.3), the predictions made by the models already developed are displayed in the news article under analysis. In Figure 3.4, the predictions made by the models developed during this thesis are presented. There are two different representations being displayed in this figure: sentences that are or not underlined and, for the sentences that are underlined there are pieces of text in different colors. An underlined sentence means that it is being predicted as a sentence containing argumentative content by the model developed in Section 4.1. On the other hand, if a sentence is not underlined it means that it is being predicted as a sentence that does not contain argumentative content by the model developed in Section 4.1. For an underlined sentence, changes in color represent the boundaries of different ADUs (given the formalism used in this thesis, ADUs can be classified between premise and conclusion). It is important to notice that each color is not associated to one specific ADU class (premise or conclusion), they are simply used in alternation to represent the boundaries of each detected ADU. The predictions related to ADU boundaries are made by the model presented in Section 4.3.

In this chapter a detailed description of the *ArgMine Framework* was presented. This framework is composed of a set of components arranged in two high-level modules: *Corpus Creation Module* and *Machine Learning Module*.

The *Corpus Creation Module* encapsulates all the components and artifacts related with the

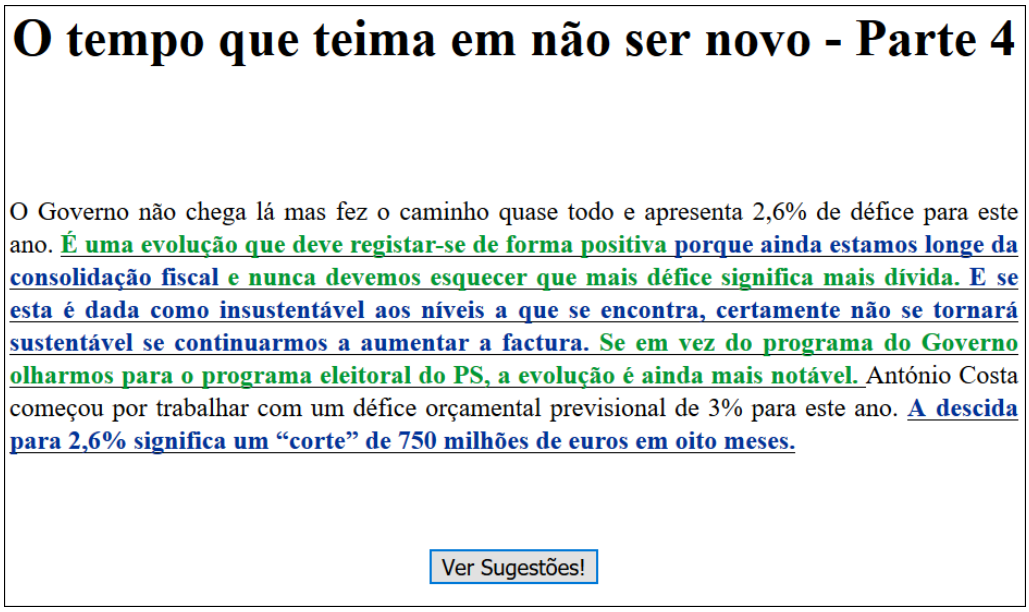


Figure 3.4: Opinion article annotated with arguments obtained from the suggestions given by the models developed in this thesis

creation of an annotated corpus with arguments from news articles written in the Portuguese language, namely *Annotation Platform* and *ArgMine Corpus*.

The *Machine Learning Module* is composed of a set components with the aim to semi-automate the process of selection and experimentation of different models and relevant features in different subtasks of the argumentation mining process.

Next, in Chapter 4, we will use the ArgMine Framework to address the first two subtasks of the argumentation mining process. Each component of the Machine Learning Module, will be instantiated according to the task that is being addressed.

# Chapter 4

# Models

As described in Section 2.3, Peldszus and Stede [PS15] divided the argumentation mining process into several subtasks, namely: text segmentation, identification of argumentative discourse units, argumentative discourse unit type classification, relation identification and relation type classification. Each subtask addresses one specific problem in the whole argumentation mining process. In this chapter, we present the proposed approaches to address the first two subtasks of the argumentation mining process. In order to build an argumentation mining system that could perform these subtasks automatically, supervised and semi-supervised machine learning algorithms will be used and properly described. Supervised machine learning algorithms create a model that maps inputs to outputs given a set of input and desired output pairs (corpus). Semi-supervised machine learning algorithms, in addition to learn from an annotated corpus, are able to make use of additional unlabeled data to better generalize the models to unseen data. In both cases, the obtained models can be used to make predictions for unseen data. Since different algorithms will be used to address different subtasks, a description of the algorithms is made in appropriate sections.

Each subtask is described in a separate section. From the models obtained in this chapter, we expect to combine their predictive capabilities to detect the arguments and identify the exact boundaries of the corresponding argumentative discourse units from news articles (namely, opinion articles) written in the Portuguese language.

The formulation and proposed approach to address the first subtask are presented in Section 4.1. In this section an assumption is made in relation to the elementary unit of analysis: we analyze the argumentative content contained in a text document at sentence level. Then, a binary classifier is trained to detect *argumentative sentences* from free text. The classifier obtained from this section is used to detect the sentences of the text document that contain argumentative content.

From the analysis of the obtained results in Section 4.1, a study of the role of argumentative keywords in text documents was performed and is presented in Section 4.2.

Finally, the formulation and proposed approach to address the second subtask of the argumentation mining process is presented in Section 4.3. In this section, a classifier is trained to identify

the boundaries of the argumentative discourse units in sentences that contain argumentative content. The obtained results are presented and critically analyzed.

## 4.1 Argumentative Sentence Detection

In this section, a description of the proposed approach to address the first subtask of the argumentation mining process is presented. In this subtask we aim to determine the zones of the input text document that contain argumentative content. To address this problem we have to define what is the desired granularity of the zones that we aim to determine as containing argumentative content or not and, what should be considered as argumentative content taking into consideration the definitions presented in Section 2.1 and the argument model used in this thesis. In relation to the granularity of the zones, we make the simplifying assumption that the elementary unit of analysis is a sentence. In relation to the definition of argumentative content, we consider a span of text as argumentative if it contains a complete argument or part of it (at least one premise or conclusion). Consequently, in this subtask, we aim to detect the sentences of the input text document that contain an argument (or part of it), which are defined as argumentative sentences.

The task of argumentative sentence detection is formulated as a binary classification problem, in which a classifier is trained on a set of annotated arguments to learn how to classify automatically each sentence as argumentative or not argumentative.

The data used for this task was obtained from the ArgMine corpus, $C^{argmine}$. A detailed description of this corpus is given in Section 3.2. For each news article $a_i$, where $a_i \in C^{argmine}$, we divided $a_i$ into sentences using the tool *Citius Tagger* [GG15], which offers the functionality of dividing a given text in different sentences as part of the process of part-of-speech tagging. Concatenating all the sentences obtained from each article $a_i \in C^{argmine}$, we obtain the dataset $X$ that will be used for the task of argumentative sentence detection. For each sentence $x_j \in X$, we determine the corresponding target value $y_j \in Y$, where $Y$ represents the set of target values, by performing the following procedure: consider the news article $a_i$, where $x_j \in a_i$, and, consider $Z$ as the set of ADUs annotated for the news article $a_i$. We consider that sentence $x_j$ has argumentative content ($y_j = 1$) if $\exists z_i \in Z : (z_i \subseteq x_j)$ *or* $(x_j \subseteq z_i)$. Otherwise, we consider that sentence $x_j$ has no argumentative content ($y_j = 0$). In Figure 4.2, an example of the instances used in this task is shown. Sentences that are underlined correspond to sentences that have argumentative content. The remaining sentences correspond to sentences that have no argumentative content.

The transformation of the annotations available in $C^{argmine}$ to learning instances previously described, corresponds to the instantiation of the *Data Preparation Component*, described in Section 3.3, for the task of *Argumentative Sentence Detection*. Given the version of the ArgMine Corpus, $C^{argmine}$, considered for the experiments presented in this thesis, in the end of the *Data Preparation Component*, we obtained a total of 360 instances (147 argumentative sentences and 213 non argumentative sentences), which were extracted from 50 documents.

Even for humans, the detection and identification of the arguments contained in a text document is considered a sophisticated skill and requires training. Often, humans rely on their cognitive capabilities to identify and understand arguments based on the context, knowledge about the world and common sense. But sometimes the task of identifying the arguments contained in text can be made easier by the presence of certain premise and conclusion indicators. These lexical cues (keywords), often expressed in the form of conjunctions or certain kinds of adverbial groups, when present in argumentative sentences directly indicate the structure of the argument. For instance, if a statement is made and preceded by the word “because", then it is quite likely that the first statement is presented as a conclusion, supported by the statement(s) that proceeds the word “because" in the corresponding text (e.g. sentence (a) in Figure 4.1). However, in some situations, this lexical cues can be misleading. In this situations, words that are typically associated with explicit argumentative content, are used in a sentence with other purpose. The ambiguity of natural language text is one of the major challenges that argumentation mining systems have to face.

Moreover, another challenge that argumentation mining systems have to face is the lack of lexical cues that directly indicate the presence of arguments. In such cases, other cues have to be explored in order to detect the arguments that are being made, such as, semantic or world knowledge, topic modeling, changes in verb tense, sentiment detectors (often associated to conclusions), named entity disambiguation. The example depicted in Figure 4.1 clearly shows that the same argument can be expressed using different written styles, which can lead to the presentation of the same argument in different ways. In (a) the argument is made explicit with the presence of the connective word “because". In (b) the relation of support between the premise and conclusion is implicit, even if the same argument, as the one presented in (a), is being expressed. In (b), the cues that can be used to identify the argument are changes in verb tense (i.e. change in verb tense from “appeared" to “had gone", which indicates the temporal ordering of the events) and causal relationship between origin of books and the consequences of insolvencies (which require world knowledge).

In order to address the problem of *Argumentative Sentence Detection* taking into consideration the different characteristics that the arguments contained in pieces of natural language texts may have, we considered that the set of tokens relevant to detect and identify the argumentative content expressed in a sentence can be divided into two categories:

- argumentative keywords: clue words that directly indicate the presence and structure of the argument contained in a span of text. Typically, these words are not domain specific. Some determinants, conjunctions, interjections, adverbial groups and verbs are the most commonly used classes of words in this category;

- domain words: in valid arguments, the conclusion needs to be supported by suitable inferences from terms provided in the premises. One cannot conclude something about “X", unless “X" is given in one or other of the premises [Dav12]. Therefore, semantically similar words are expect to be found in the different argument components. Typically, these words

are domain specific. Nouns, named entities, pronouns, adjectives and verbs are the most commonly used classes of words in this category.

The categorization of the tokens contained in a sentence between argumentative keywords and domain words, will be useful to better understand some of the features that are employed to address the task of *Argumentative Sentence Detection*, which will be described in Section 4.1.2.

(a) The book never appeared, because the publisher had gone bankrupt.

(b) The book never appeared. The publisher had gone bankrupt.

Figure 4.1: Two different writing styles expressing the same argument [PS13]

### 4.1.1 Algorithms

The task of Argumentative Sentence Detection was formulated as a binary classification problem, in which we aim to classify each sentence as argumentative sentence or non argumentative sentence.

In this subsection, we instantiate the *Machine Learning Algorithms and Techniques* component from the *ArgMine Framework*, according to the approach made to address the current task. In relation to machine learning algorithms, experiments were made using the following classifiers: *naïve Bayes classifier*, *Maximum Entropy Model*, *Support Vector Machine*, *Random Forest Tree* classifier and *Stochastic Gradient Descent* algorithm. In order to reduce the dimensionality of the feature space, we performed experiments using feature extraction techniques (*PCA* and *LDA*) and feature selection techniques (*variance threshold* and *percentile selection*).

### 4.1.2 Features

In order to apply machine learning algorithms it is necessary to represent the training instances by a set of numerical features. A good set of features should represent the training instances in such a way that would make it possible for the machine learning algorithms to find patterns in the data which can be used to classify instances according to the desired target labels. For the task of Argumentative Sentence Detection, the training instances are sentences that occur in a text document, which should be classified as argumentative sentence or not argumentative sentence. Each sentence is represented with a set of features at the lexical, syntactic and semantic level:

- N-Gram: contiguous sequence of 1 to $N$ tokens from a given sentence. This feature was used as a baseline to compare with more specific features. We encode the presence of unigrams, bigrams, and trigrams in the sentence ($N = 1$, $N = 2$ and $N = 3$, respectively);

Um elemento químico é uma substância que contém apenas um único tipo de átomos. O nome de cada elemento é abreviado em uma ou duas letras, que constituem o seu símbolo químico. Uma grande proporção de elementos são metais. **Os átomos de um mesmo elemento não têm todos o mesmo peso, pois alguns contêm mais neutrões.** Estes elementos químicos são organizados na TP em função das semelhanças e das diferenças que existem entre eles.

Figure 4.2: Opinion article annotated with one argument

- Word couples: all possible combinations of word pairs within a sentence. Using this feature, we expect to retrieve pairs of words that capture argumentative reasoning, appearing not necessarily adjacent to each other. These pairs of words occur typically together in the same sentence and are often associated to argumentative content. Since the pair of words are not necessarily adjacent to each other, this feature increases the feature space substantially. For this reason, we also did experiments with a cleaned corpus, in which all the punctuation marks, numbers and nouns were removed (e.g.“*Concluo [...] porque [...]"* (“I conclude [...] because [...]"), “*Se [...] então [...]*" (“If [...] then [...]"));

- Argumentative keywords: set of clue words directly indicating the structure of the argument. These words are strong indicators of argumentative content. A set of argumentative keywords, $K$, that are typically found in argumentative text written in Portuguese was manually compiled, based on the work presented in [Coh84]. The set of argumentative keywords $K$ contains a total of 51 argumentative keywords (e.g. *“logo"* (“thus"), *“pois"* (“because"), *“portanto"* (“therefore")). This feature is encoded as a binary feature: if the sentence contains at least one word which belongs to the set of argumentative keywords $K$ then, the feature is set to 1; otherwise, the feature is set to 0 (e.g. in the underlined sentence shown in Figure 4.2 this feature will be set to 1 due to the presence of the word *“pois"*);

- Text statistics:

  - Absolute Position: current sentence absolute position in relation to the document where the sentence was extracted (e.g. for the underlined sentence in Figure 4.2 - 3);
  - Average Word Length: words used in argumentative sentences might have different characteristics from words used in non argumentative sentences. This feature explores if this difference occurs in the average length of the words (e.g. for the underlined sentence in Figure 4.2 - 4.0);
  - Number of punctuation marks: argumentative sentences may increase the number of punctuation marks in the sentence (e.g. for the underlined sentence in Figure 4.2 - 1);
  - Sentence Length: number of words in current sentence (e.g. for the underlined sentence in Figure 4.2 - 19);

- Adverbs: some adverbs can signal argumentative content (e.g. *"então"* ("so"), *"sempre"* ("always"), *"mas"* ("but"), amongst others);

- Modal Auxiliary: words indicating the level of necessity, which are usually found in some types of arguments (e.g. *"poder"* ("can"), *"dever"* ("must"), *"ter"* ("have"), amongst others);

- Verb tense: changes in verb tense can often be found in argumentative context. For instance, arguing about something in the present supported by premises that occurred in the past. Given a sentence $s_i$ we explored changes in the verb tense that occur in the sentence $s_i$ and, between the sentence $s_i$ and the surrounding sentences, $s_{i-1}$ and $s_{i+1}$ (e.g. in the sentence (b) from Figure 4.1, changes in verb tense between "appeared" and "had gone" indicate a sequence of events which, in some situations, are associated to argumentative content). A window size of length 3 (current, previous and next sentences) was considered in this feature due to the assumption that the ADUs must occur in sequential spans of text and, therefore, analyzing sentences that are not in the neighborhood is not necessary. When analyzing changes in verb tense between different sentences, we consider the verbs that are closer to each other. A change in verb tense between two sentences, $s_i$ and $s_{i-1}$, occurs if the last verb not in the infinitive form from sentence $s_{i-1}$ has a different verb tense than the first verb not in the infinitive form from sentence $s_i$. A change in verb tense between two sentences, $s_i$ and $s_{i+1}$, occurs if the first verb not in the infinitive form from sentence $s_{i+1}$ has a different verb tense than the last verb not in the infinitive form from sentence $s_i$. The information related to verb tenses is obtained from the part-of-speech tool *Citius Tagger* [GG15], which classifies each verb with one of the following verb tense categories: Present, Imperfect, Future, Past or Conditional;

- Domain words repetition: arguments have to be about something and, therefore, repetitions of domain words or the existence of similar domain words are expected in different components of the argument. In this feature repetitions of nouns, name entities, verbs and adjectives were considered. All the punctuation marks and discourse markers were removed in the cleaning process. Given a sentence $s_i$ we explored word repetitions occurring in the sentence $s_i$ and, between sentence $s_i$ and the surrounding sentences, $s_{i-1}$ and $s_{i+1}$. A window size of length 3 (current, previous and next sentences) was considered in this feature due to the same reason explained in the previous feature. Using an word embeddings model generated for the Portuguese language [ARPS13], we calculate the similarity between two words using the metric cosine similarity between the word feature vectors that represent each of the words. We calculate the similarity between each pair of words occurring in sentence $s_i$, between pairs of words in sentence $s_i$ and sentence $s_{i-1}$, and between pairs of words in sentence $s_i$ and sentence $s_{i+1}$, separately. For each of them, the similarity score of the most similar pair of words is encoded directly as a feature (e.g. in the underlined sentence from Figure 4.2, this feature should capture the similarity between the words "átomo", "elemento" and "neutrões", which correspond to similar words related to the topic of the argument; in

the sentences from Figure 4.1 the most similar words are “book" and “publisher", which are also related to the topic of the argument that is being presented).

To scale and normalize the mentioned set of features, the *tf-idf* method was used to scale each set of features that is based on a vocabulary of words (*N-Gram*, *Word couples*, *Modal auxiliary*, *Adverbs*), and all numerical features are scaled to a range between 0 and 1, using the method *MinMaxScaler* provided by *scikit-learn* [PVG+11].

The *tf-idf* representation, short for *term frequency-inverse document frequency* representation, is a weighting scheme commonly used to scale features based on a vocabulary of words. Term frequency (*tf*) measures the raw frequency of a term in a document (i.e. the number of times that a term $t$ occurs in a document). Inverse document frequency (*idf*) is a measure of how much information the word provides, that is, whether the term is common (low *idf* score) or rare (high *idf* score) across all documents. Combining both measures, we obtain the *tf-idf* measure. An high *tf-idf* value is reached by a high term frequency in a given document and a low document frequency of the term in the whole collection of documents.

As previously described, in the *Domain words repetition* feature, we exploit a distributed representation of words (word embeddings). These distributions map a word from a dictionary to a feature vector in high-dimensional space, without human intervention, from observing the usage of words on large (non-annotated) corpora. This real valued vector representation tries to arrange words with similar meanings close to each other based on the occurrences of these words in a corpora. Then, from these representations, interesting features can be explored, such as semantic and syntactic similarities. In the experiments presented in this thesis, we used a model provided by the tool *Polyglot*[1], in which a neural network architecture was trained on Portuguese *Wikipedia* articles. A full description of the tool can be found in [ARPS13]. In order to obtain a score indicating the similarity between two words, we compute the cosine similarity between the vectors that represent each of the desired words in the high-dimensional space.

### 4.1.3 Results

The results presented in this section were obtained from experiments made with the *ArgMine Framework*.

The best results were obtained using a *Support Vector Machine* classifier with a linear kernel. From all the features described in Section 4.1.2, the following subset of features yield the best results: word couples, argumentative keywords, average word length, absolute position, modal auxiliary, adverbs, verb tense and domain words repetition.

In order to reduce the dimensionality of the feature space and to remove some of the feature values that are constant or have low variability across the dataset, feature analysis techniques were applied. For this task, we obtained better results using the *Variance Threshold* method: all features whose variance does not meet some threshold are removed from the feature set. The best results were obtained using a threshold of 0.0001.

[1] http://polyglot.readthedocs.io/en/latest/index.html

The results depicted in Table 4.1 were obtained in a five-fold cross-validation scenario.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| no argument | 0.70 | 0.68 | 0.69 | 213 |
| argument | 0.56 | 0.59 | 0.57 | 147 |
| avg / total | 0.64 | 0.64 | 0.64 | 360 |

Table 4.1: Argumentative Sentence Detection Scores

For every type of label and for the overall performance measure, the precision, recall and f1-score were used as evaluation metrics. Precision is the measure of what percentage of predicted labels were correctly classified. Recall is the measure of what percentage of the labels in the gold-standard dataset were correctly classified. Finally, f1-score is the harmonic mean of precision and recall. These metrics are commonly used in classification systems to evaluate their performance.

We obtained better overall results in the detection of non argumentative sentences (0.69), as compared to the results obtained in the detection of argumentative sentences (0.57), which we associate with the higher number of non argumentative sentences contained in the *ArgMine Corpus*.

### 4.1.4 Error Analysis

In this section, we critically analyze the errors made by the system.

Unlike what could be expected, the presence of lexical clues (argumentative keywords) does not seem to be as relevant as we initially thought for the detection of argumentative sentences. Some of the lexical clues, that are typically associated with explicit argumentative content, are often found also in non-argumentative sentences, transforming this intuitive set of features into an irrelevant set of features for the classifiers. As shown in Figure 4.3, the lexical clue *"logo"*, which is typically associated to argumentative content, is contained in a non argumentative sentence. Conversely, argumentative sentences do not necessarily contain such clues. For instance, in the sentences depicted in Figure 4.4, there is no lexical clue indicating the presence of the argument and an interpretation of the content at a contextual level is necessary to identify the argument.

Cavaco desconcertou, é certo. Mais que qualquer outra coisa, desconcertou e confundiu. Tinha uma diferente escala de prioridades, veio com outra carta de intenções. Pertencia a outro mundo. Com as vantagens que isso pode ter, com as desvantagens que tal risco comporta. Começou **logo** na campanha eleitoral para as legislativas de 1985 que ganhou desajeitadamente, seduzindo uns, incomodando outros mas espantando todos.

Figure 4.3: Example of lexical clue contained in non argumentative sentence

Also, we observe that some features tend to be problematic in terms of overfitting. Feature sets based on dictionary of words, such as *N-Gram*, *Word Couples* and *Adverbs*, create big and sparse sets of individual features (typically, each token in the training set is represented by a feature), in which some of the individual features are considered as relevant by the classifier even though the

Os quatro países vão reforçar a cooperação nas questões da imigração, terrorismo e tráfico de droga. Esta manhã foram assinados vários acordos nestas áreas, numa reunião que decorreu no Forte de São Julião da Barra em Oeiras, e que contou com os ministros da Administração Interna dos quatros países. Um dos temas mais sensíveis foi o da gestão dos fluxos migratórios que passa por um reforço de cooperação em matéria de gestão e controlo de fronteiras.

Figure 4.4: Example of argumentative sentences without lexical clues

token itself, at the lexical level, should not be considered as indicative of argumentative content. This situation occurs with tokens that are domain specific and that appear only, or mostly, in argumentative sentences. Even if this situation should mean that the token is a good indicator of argumentative content (as it corresponds to the intuition given by the *tf-idf* representation), there are some characteristics of the ArgMine corpus that seem to transform this intuitive set of features into an irrelevant set of features: the limited size of the ArgMine corpus and the variability of topics that are covered in different articles. The other category of features that are being problematic in terms of overfitting is related to the *Absolute position* feature. This feature is considered relevant in the sense that the classifier learned that sentences occurring in the begin of the document have higher probability of being argumentative sentences. From the analysis made to the annotations contained in the ArgMine Corpus, we verified that the majority of the argumentative sentences occur in the beginning of the document, fact that explains this outcome. However, the importance assigned by the classifier to this feature indicates that the quantity of annotations presented in our dataset is too small.

In sum, our results can be explained in two dimensions. On one hand, a detailed analysis of the features deemed as relevant by the classifier clearly indicate that our corpus is too small. Moreover, given the aforementioned lack of relevance of lexical clues, we conclude that lexical and syntactic-based approaches are not enough to address this complex task of identifying argumentative sentences, as the example depicted in Figure 4.4 clearly shows.

### 4.1.5 Conclusions

In this section, a description of the proposed approach to address the task of Argumentative Sentence Detection, which corresponds to the first subtask of the argumentation mining process, was presented.

The detection and identification of arguments contained in a text document is complex, and several challenges have to be overcome, such as: the ambiguity of natural language text, lack of clues directly indicating the structure of the arguments, amongst others.

The task was formulated as a binary classification problem and a support vector machine classifier was trained to automatically perform this task from the annotated arguments contained in the ArgMine Corpus. From a detailed analysis of the obtained results, we conclude that the reduced number of annotations available in the ArgMine Corpus and, in addition, the variability of topics that are covered in different articles make the current task even more challenging to be addressed

using machine learning algorithms. Being argumentation mining a complex task, the quantity and quality of annotations is crucial and should be significantly higher in order to avoid problems related to overfitting and, in order to make the learning process feasible. One important characteristic of the ArgMine corpus is the variability of the topics that are covered in different news articles. Many approaches presented in the state-of-the-art (namely in Section 2.4) make strong use of the knowledge of the specific application domain in which they work. For instance, the work developed by Mochales and Moens [PM09] on legal documents, make use of the presence of specific syntactical descriptors or keywords, as they are frequent in this type of documents, and provide reliable clues for the detection of recurrent patterns that can be useful for the task of Argumentative Sentence Detection. Working with a corpus that is not domain specific poses many challenges and requires a sophisticated set of features to be explored in order to find reliable patterns in the data. For instance, the exploration of semantic-level features that could be useful in the task of Argumentative Sentence Detection seems to be a crucial step forward to improve argumentation mining systems across different topics and application scenarios.

Another important consideration made in this section is related to the granularity of the zones of the text in which we aim to determine the argumentative content. We made the assumption that the elementary unit of analysis is a sentence. Due to the characteristics of the arguments presented in text documents, this assumption can have consequences in terms of the feasibility of the learning process: since sentences may contain complete arguments or part of arguments (i.e. premises and/or conclusion), this implies that the learning instances may have some variability in the argumentative content that they contain. This consideration is particularly relevant taking into account the characteristics of the news articles contained in the ArgMine Corpus. From the analysis of the annotations contained in the corpus, we conclude that the ADU's are typically sentence level or smaller spans of text. Therefore, different approaches to this problem may be more adequate. In Section 4.3, we approach the granularity of the arguments in more detail.

From the analysis of the features deemed as relevant by the classifier we conclude that the ArgMine corpus is too small and, a significantly greater number of annotations is required. Besides this, some improvements to the implemented features and more sophisticated features should be applied in order to improve the results of the system in the task of Argumentative Sentence Detection. For instance, in the *Domain words repetition* feature set, the approach made to address the repetition of named entities and pronouns is simplistic and improvements in this specific task may yield significant improvements in the task of Argumentative Sentence Detection.

The predictive capabilities of the model developed in this section are being applied to suggest, in unannotated texts, potential arguments to users in the *Annotation Platform* (Section 3.1).

## 4.2 The role of keywords

In the previous section, we addressed the problem of Argumentative Sentence Detection, which aims to determine the sentences in a text document that have argumentative content. We consider a sentence as argumentative if it contains a complete argument or part of it (at least one premise or

conclusion). The task is formulated as a binary classification problem, in which a *Support Vector Machine* classifier was trained on a set of annotated arguments available in the *ArgMine Corpus*. Machine learning algorithms, such as *Support Vector Machines*, require a set of numerical features to represent the training instances. For the task of *Argumentative Sentence Detection* each sentence is represented with a set of features at the lexical, syntactical and semantic level, as described in Section 4.1.2. One of the implemented features, the *Argumentative Keywords* feature, is a binary feature indicating whether the sentence contains a word from a pre-determined set of words that are considered as argumentative keywords. This set of words was manually compiled, based on the work presented in [Coh84]. Argumentative keywords are words that are typically found in argumentative texts and whose presence directly indicates the structure of the argument.

From the critical analysis of the obtained results in the Argumentative Sentence Detection task, discussed in Section 4.1.4, we concluded that the ambiguity associated to the lexical clues transformed this intuitive set of features into an irrelevant set of features for the classifiers. In some situations, considering a word as an argumentative keyword just taking into account lexical information can induce some errors. For instance, we can erroneously indicate that a word is an argumentative keyword in situations where it is clear that the word under consideration has no argumentative role. As shown in Figure 4.3, the word *"logo"*, typically associated to argumentative content, is considered argumentative keyword even though it is contained in a non argumentative sentence. As a consequence, considering a word as argumentative keyword taking into account only lexical information is too simplistic and can have some drawbacks when used in a machine learning problem. Following this line of reasoning, we investigated if there was some additional property that should be verified to consider a word as an argumentative content indicator. This property can be at the word level or, related to the role of the word taking into account the whole sentence (at sentence level).

We explored if such an additional property could be related to the syntactic role of a given word. To verify this hypothesis, we performed a different approach to construct the list of argumentative keywords. Instead of a manually compiled list of keywords, which does not take into account the characteristics of the target texts used in the argumentation mining process, we did the following procedure: starting from a list containing all the words in the Portuguese language, obtained from the Portuguese dictionary provided by *Jspell* [AP94], we removed all the words that are not domain specific (we considered adverbs, conjunctions, determiners and verbs) and all the words that are not used in the *ArgMine Corpus*. After this procedure, we obtain a new set of candidate argumentative keywords. The feature set is obtained considering the syntactic information of each individual token and from the syntactic information of the surrounding tokens in the sentence under analysis. Therefore, assuming that the number of words in the dictionary is $n$, and knowing that each word has three features associated (the syntactic information of the current word and, the syntactic information of the previous and following word occurring in the sentence), the total number of features in the feature space will be $3 * n$ .

Using the feature set previously described, we employed a *Support Vector Machine* with a linear kernel applied to the task of *Argumentative Sentence Detection*: using this setting and, from

the analysis of the features deemed as relevant by the classifier, we expected to be able to draw some conclusions in relation to the words and corresponding syntactic information that are more useful when addressing the task of *Argumentative Sentence Detection*.

However, due to the sparsity of the obtained feature space and due to the low number of training examples available in the ArgMine Corpus, the analysis of the obtained results indicate that the model is giving importance to words that are corpus-specific and, therefore, the obtained model is not able to generalize well. For this reason, we conclude that the problem formulated in this section is infeasible given the available resources.

## 4.3 ADU Boundary Detection

The aim of *ADU Boundary Detection* is to identify the boundaries of the argumentative discourse units (ADUs) presented in the text. As previously defined, an ADU is the elementary unit of the arguments contained in the text and can be classified between premise and conclusion given the formalism used in this thesis. In this section, we formalize the problem of *ADU Boundary Detection* and a description of the approach used to address this problem is presented.

The task of *ADU Boundary Detection* corresponds to the second step in the *Argumentation Mining Process*, as described in Section 2.3. Comparing with the previous step in the process, which was addressed in Section 4.1, this step implies the study of the segmentation problem in more detail and at a granularity level more suited to the argumentation mining concepts introduced in Section 2.1. In this section, a deeper analysis of the argumentative content contained in a text is performed: besides the detection of the sentences that contain argumentative content, we aim to identify the exact boundaries of the ADUs (i.e. premises or conclusions).

As discussed in [PM09], the choice of the granularity level from which the segmentation problem has to be addressed is highly dependent on the type of text at hand. More formal texts, such as legal documents, “present longer sentences with many subordinate sentences. Therefore, these texts present premise and conclusion in subordinate sentences or independent sentences instead of subclauses" [PM09, p. 101]. More informal texts, such as news articles, “will contain shorter sentences where conclusion and premise can be together in a single sentence, being each a subclause of the sentence" [PM09, p. 101]. Since the type of texts contained in the *ArgMine corpus*, $C^{argmine}$, are news articles and from the manual analysis of the granularity level used to annotate the ADUs in the annotations that were collected, we conclude that a deeper analysis on smaller text spans should be formalized in order to successfully address the problem of *ADU Boundary Detection*. Even if, in some few situations, the ADUs annotated in $C^{argmine}$ contain a piece of text from more than one sentence, the majority of the annotated ADUs are sentence level or smaller spans of text. Therefore, the problem of *ADU Boundary Detection*, has to be considered as a segmentation problem in which the ADU boundaries can be spans of text smaller than sentences. For this reason, the assumption made in Section 4.1, in which we assumed that the elementary unit of analysis are sentences, is too strong and restrictive for the task that will be addressed in this section. In order to cover all the possibilities, we assume that all ADUs containing more than one

sentence can be decomposed into several sentence level (or smaller spans of text) ADUs, which can be linked together without loosing expressiveness in terms of argumentative content.

Analyzing a text source to find element boundaries at word level is far more complex than an analysis at sentence level. Since the complexity of the task increases, the number and quality of annotations that are required to address the problem should also increase. Even if the *ArgMine Corpus* is being annotated with the necessary information to address the task of *ADU Boundary Detection*, the low number of annotations that is available at the time of this writing makes the task infeasible if it is based only on the knowledge that is possible to extract from the *ArgMine Corpus*.

Thus, a two step approach is used to address the task of *ADU Boundary Detection*. The first step, *Argumentative Sentence Detection*, was formalized and a proposed solution was presented in Section 4.1. From this step, we obtain the sentences of the original text that are predicted as containing argumentative content and, consequently, they correspond to the zones of text where a deeper analysis is necessary to be made in order to retrieve the desired ADUs boundaries. In the second step, which will be formalized in this section, a classifier is trained using an external corpus with the aim of learning how to divide a complete sentence into small spans of text, which correspond to elementary discourse units (EDUs) as defined by the RST framework. From this two step approach the following assumption is made: *given an argumentative sentence, ADUs boundaries correspond to EDUs boundaries, where EDUs boundaries are defined as the convention used by the RST framework*.

The assumption that will be studied in this section, was motivated by the existence of a corpus containing news articles written in the Portuguese (Brazilian) language which are annotated using the *RST* framework. *RST* is a theory of discourse closely related to some of the concepts of Argumentation Mining, as described in Section 2.2. Some researchers in argumentation mining use this theory of discourse as the basis of their work. Being a theory of discourse, the annotations are rich in terms of discriminating the discourse markers presented in the text. This is interesting in the point of view of learning the ADUs boundaries because it is expected that some groups of discourse markers play an important role as indicators of ADU boundaries.

The data used for this task was obtained from CSTNews corpus [CMJ+11], henceforth $C^{rst}$, which contains news articles annotated according to the RST framework. Each annotation contains two files: a text file containing the complete news article and a file containing the RST tree of the corresponding news article. For each news article $a_i \in C^{rst}$, we divided $a_i$ into sentences using the tool *Citius Tagger*. Performing this preprocessing step is in accordance with the problem formulation because sentence boundaries are always EDU boundaries (as defined in the RST framework) and, therefore, no information is lost for the task of boundary detection if instead of working with the complete text document as input we work at the sentence level. On the other hand, using an external tool to perform sentence level segmentation facilitates the current task: instead of receiving as input the original text without any segmentation information, we can receive as input the original text segmented at a sentence level. Therefore, we can focus on finding intra-sentence EDU boundaries instead of accumulating the additional problem of performing sentence

segmentation. Concatenating all the sentences obtained from each article $a_i \in C^{rst}$, we obtain the dataset $X$ that corresponds to the input of the current task. Each sentence $x_j \in X$ is represented by a set of tokens ($x_j = (t_0, ..., t_n)$), where $n$ corresponds to the number of tokens contained in the sentence $j$, which were obtained using the tool *Citius Tagger*.

The set of labels $Y$ is represented with *BIO encoding*. Each token is labeled with one of the 4 categories: *B-VP* (Begin of Verb Phrase), *I-VP* (Intermediate of Verb Phrase), *B-EDU* (Begin of EDU) or *O* (Other).

**Original Sentence:** *"A posição de Quintanilha tem o apoio de alguns senadores, mas não é bem vista pela oposição."*

**Training Example:**

| $t$ | $y_t$ | $x_t$ |
|---|---|---|
| 0 | B-EDU | A |
| 1 | O | posição |
| 2 | O | de |
| 3 | O | Quintanilha |
| 4 | B-VP | tem |
| 5 | I-VP | o |
| 6 | I-VP | apoio |
| 7 | I-VP | de |
| 8 | I-VP | alguns |
| 9 | I-VP | senadores |
| 10 | I-VP | , |
| 11 | B-EDU | mas |
| 12 | O | não |
| 13 | B-VP | é |
| 14 | I-VP | bem |
| 15 | B-VP | vista |
| 16 | I-VP | pela |
| 17 | I-VP | oposição |
| 18 | I-VP | . |

Figure 4.5: One example of a training instance extracted from CSTNews corpus for the task of EDU Boundary Detection

From $C^{rst}$, we obtain news articles segmented in EDUs. Each news has a corresponding XML file with the annotation content following the format defined in the RST Framework. In these files, each EDU corresponds to a *"segment"* element and RST relations between adjacent EDUs are indicated in the *"relname"* attribute of each element. To transform this resource into relevant data for the task of *EDU Boundary Detection*, it is necessary to have the original text annotated with EDU boundaries. By simply concatenating the content from all the *"segment"* elements in the

annotation file it is possible to obtain the desired information because the elements that appear in the annotation file are in the same order in which they occur in the original text. Consequently, the assignment of labels to each token from a news article is performed with the following procedure: first, the beginning of each EDU is labeled with *B-EDU*. For the remaining tokens, if the token in position $k$, where $k \in [2, m]$ and $m$ corresponds to the length of the EDU, is a verb then it receives the label *B-VP*. If the token does not correspond to the beginning of an EDU and it occurs after a token labeled with *B-VP*, then it receives the label *I-VP*. The remaining tokens receive the label *O*. An example of applying this labeling procedure to one of the instances of the dataset is depicted in Figure 4.5.

In order to address the task of *EDU Boundary Detection*, the useful information that can be obtained in the annotation files from $C^{rst}$ is the token that is in the beginning of an EDU, which is labeled as B-EDU in this formulation. However, using only this information to identify the boundaries of the EDUs often results in boundaries that do not take into account some of the characteristics that the resulting EDUs must verify. As introduced in Section 2.2, in order to consider a piece of text as an EDU, annotators have to follow a set of rules that were defined in the RST framework. One of the main restrictions that a piece of text must follow to be considered an EDU is that it must contain at least one verb. This restriction is also relevant for the task of *ADU Boundary Detection*: as defined in Section 2.3, an ADU is a proposition that can be classified between premise or conclusion depending on the argumentative role that it has in the text and, therefore, in order to consider a piece of text as an ADU the minimal condition that we can apply is that it must contain a verb. Consequently, this restriction must be included in the formulation of the problem.

As introduced in Section 4.3.1, the algorithm *Conditional Random Fields* optimizes the sequence of labels globally. Therefore, in order to employ the restriction that a B-EDU should be between two spans of text containing at least one verb, we added the labels B-VP and I-VP to the formulation of the current task, as previously described. A token labeled with B-VP means that the corresponding token is a verb and, a token labeled with I-VP means that the corresponding token is not a verb but that one of the previous tokens in the sequence is a verb. The combination of tokens B-VP and I-VP indicate the presence of a piece of text that contain at least one verb.

Using a Linear-chain Conditional Random Field, we aim to build a sequential classifier that learns to identify the boundaries of the EDUs. With the formulation used for this task, we expect that the classifier learns that a B-EDU must occur after a sequence of tokens labeled with B-VP or I-VP and, that a valid sequence of labels should always finish with B-VP or I-VP.

Since the part-of-speech of each token in the sequence is known, then the label B-VP should be easy to classify. Therefore, we expect to obtain very high scores for the label B-VP. It is important to notice that the purpose of adding the labels B-VP and I-VP is to represent better the data for the task of *EDU Boundary Detection* and not as a target label. So, using the information of the part-of-speech directly as an emission feature will reduce the complexity of the task and facilitate the identification of valid EDUs boundaries.

### 4.3.1 Algorithms

To address the task of *EDU Boundary Detection* we employ sequential models, where the inputs are assumed to have sequential dependencies.

In this subsection, a description of the Linear-chain Conditional Random Field [LMP01, SM12] algorithm is made.

Conditional Random Fields (CRFs) are particular instances of undirected probabilistic graphical models, which have a chain topology. In a graphical model, every random variable is represented as a node in a graph, and the edges in the graph represent probabilistic dependencies between random variables. The random variables are divided into two sets, the *observed variables* (or *observations*), $x = \{x_1, x_2, ..., x_n\}$ (e.g. sequence of words from a text with length $n$), and the *hidden variables* (or *states*), $y = \{y_1, y_2, ..., y_n\}$ (e.g. sequence of states that corresponds to labels that must be assigned to each word in the input sequence $x$).

Let $G = (V, E)$ to be an undirected graph such that there is a node $v \in V$ corresponding to each of the random variables representing an element $y_i \in y$. Then $(x, y)$ is a conditional random field when the random variables $y_i$, conditioned on $x$, obey the Markov property with respect to the graph $G$.

In a linear-chain CRF the states are represented in a linear structure, in which the $i^{th}$ state $y_i$ depends only on the previous state $y_{i-1}$ (first-order Markov assumption), as represented in Figure 4.6.

Then, a linear-chain conditional random field defines the conditional probability of a state sequence given an input sequence that takes the form:

$$p(y|x) = \frac{1}{Z_x} exp(\sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_j f_j(y_{i-1}, y_i, x, i)) \tag{4.1}$$

where,

- $Z_x$ is a normalization factor of all state sequences;
- $f_j(y_{i-1}, y_i, x, i)$ is one of the $m$ feature functions that takes as input a sentence $x$, the position $i$ of a word in the sentence, the label $y_i$ of the current word, the label $y_{i-1}$ of the previous word and, outputs a real value number and;
- $\lambda_j$ is a weight assigned to each feature function $f_j$, whose value is learned from the data. Intuitively, the learned feature weight $\lambda_j$ should be positive for features correlated with the target label, negative for features anti-correlated with the label, and near zero for relatively uninformative features.

Training involves finding the $\lambda$ parameters. Since CRFs define the conditional probability $p(y|x)$, the appropriate objective for parameter learning is to maximize the conditional likelihood of the training data. However, it is not possible to analytically determine the parameter values that maximize the log-likelihood, since setting the gradient to zero and solving for $\lambda$ does not always yield

a closed form solution. Instead, the standard parameter learning approach is to use iterative techniques (such as, *iterative scaling*) or gradient-based methods (such as, *SGD* or *L-BFGS*).

Given the learned parameters and a new observation sequence $x = \{x_1, x_2, ..., x_n\}$, we want to find the sequence of labels $y^* = \{y_1^*, y_2^*, ..., y_n^*\}$, such that $y^* = argmax_y p(y|x)$. This is called the *decoding problem*. Performing such calculation in a naïve way is intractable due to the required sum over label sequences: if the observation sequence $x$ has $n$ elements and the number of different labels is $k$, there are $n^k$ possible label sequences. Summing over this number of terms is prohibitively expensive. Typically, we rely on (polynomial-time) dynamic programming algorithms to find the optimal label sequence (such as, *Viterbi algorithm*, *Forward-backward algorithm*).

In our case, we used *CRFSuite*[2] implementation of Conditional Random Field algorithm in the experiments performed in this thesis.

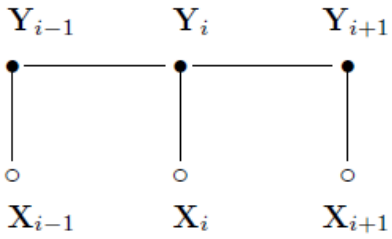


Figure 4.6: Graphical structure of linear chain CRF for sequences. An open circle indicates that the variable is not generated by the model [LMP01]

### 4.3.2 Features

To define the *linear chain CRF* for this problem, we need to choose the set $F$ of feature functions $f_j(y_{i-1}, y_i, x, i)$. As defined by Lafferty *et al.* [LMP01], we can divide the feature functions of a linear chain CRF into *transition features* (or label-label features), represented as the horizontal edges in Figure 4.6, and *emission features* (or label-word features), represented by the vertical edges in Figure 4.6.

The first kind of features, transition features, correspond to all pair-wise combinations of the labels $Y$. Since, in the formalization of this problem, we have four possible labels, then there are sixteen transition features.

The emission features defined for this problem are the following:

- Strong punctuation mark ($F_{spm}$): if $x_{i-1}$ is strong punctuation mark then return 1; otherwise return 0.
  As strong punctuation marks the following tokens were considered: '.', ';', '!' and '?';

[2] http://www.chokkan.org/software/crfsuite/

- Weak punctuation mark ($F_{wpm}$): if $x_{i-1}$ is weak punctuation mark then return 1; otherwise return 0.
  As weak punctuation marks the following tokens were considered: ',' and ':';
- Inside parenthetical text ($F_{par}$): if $x_{i_1} = \{$'(', '[', '-'$\}$ and $x_{i_2} = \{$')', ']', '-'$\}$ , $i_1 \in [1, i[$ , $i_2 \in ]i, n]$ , $n = sentence\,length$, then return 1; otherwise return 0;
- Part-of-speech [1] of the word $x_{i-1}$, $x_i$ and $x_{i+1}$ ($F_{pos_{-1}}$, $F_{pos}$ and $F_{pos_{+1}}$, respectively);
- Lemma [2] of the word $x_{i-1}$, $x_i$ and $x_{i+1}$ ($F_{lem_{-1}}$, $F_{lem}$ and $F_{lem_{+1}}$, respectively);
- Begin of sentence ($F_{bos}$): if $i == 0$ return 1; otherwise return 0;
- End of sentence ($F_{eos}$): if $i == sentence\,length$ return 1; otherwise return 0;

### 4.3.3 Results

The results presented in Table 4.2 were obtained in a five-fold cross-validation scenario and, using a *Linear-chain Conditional Random Field*.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| O | 0.96 | 0.86 | 0.91 | 11905 |
| B-EDU | 0.91 | 0.83 | 0.87 | 3218 |
| B-VP | 0.99 | 0.99 | 0.99 | 6281 |
| I-VP | 0.94 | 0.98 | 0.96 | 34139 |
| avg / total | 0.95 | 0.95 | 0.95 | 55543 |

Table 4.2: EDU Boundary Detection Scores

For every type of label and for the overall performance measure, the precision, recall and f1-score were used as evaluation metrics.

Table 4.2 shows the overall performance of our method (f1-score 0.95) and, the performance for each of the labels. The score for the label *B-VP* is very high (f1-score of 0.99). This result was expected due to the fact that one of the features used in this task, namely $F_{pos}$, directly indicates the label *B-VP*. Intuitively, the algorithm learns to predict the label B-VP when the part-of-speech of the word is verb. Comparing with the other labels, *B-EDU* has the lowest overall scores, which we associate to the fact that this label has the lowest number of examples.

### 4.3.4 Error Analysis

In this section, a critical analysis of the obtained results is presented.

The labels *I-VP* and *O* are dependent of the labels *B-VP* and *B-EDU*, respectively. It can been seen from Table 4.2 that the recall values for *I-VP* and *B-VP* are lower comparing with the recall

[1] Tags obtained from the analysis of a part-of-speech tagger, which classifies each token between 12 different classes (such as adjective, verb, amongst others)

[2] The lemma of a given word is obtained from the analysis of a part-of-speech tagger

values of *B-VP* and *B-EDU*, respectively. We associate this due to propagation of errors made after misclassification of a word as *B-VP* and *B-EDU*. For instance, when a word is misclassified as *B-VP* and, since the algorithm learned that a transition from a *B-VP* label to a *I-VP* label is very likely, then it will (with some probably and, also taking into consideration the emission probabilities) wrongly classify the following words as *I-VP*.

The confusion matrix depicted in Table 4.3 shows that approximately 98% of the missed words that should have been predicted as *O* were caused by wrongly predicting the word as *I-VP*. This is probably due to the fact that both labels are similar in nature (both are labels preceding a B label, namely: *B-EDU* and *B-VP*, respectively). Another possible cause for the O label being confused with the I-VP label is the fact that approximately 79% of the missed *B-EDU* labels were made when the system wrongly predicted the token as *I-VP*, which means that the emission features of these words where not enough to make the system predict the token correctly as *B-EDU*, and that when this misclassification occurs at some token, the preceding labels will be predicted as *I-VP* instead of *O*. Conversely, the major part of the errors made by the system when predicting labels as *O* and *B-EDU* were made when the truth-value is *I-VP*. This occurs when the system wrongly classifies a word as *B-EDU* and, then, the error is propagated to the *O* label.

From a manual analysis of the predictions made by the system, we were able to identify some words that are ambiguous for this task, namely: *"que"*, *"e"*, *"para"*, *"como"*. For instance, the word "que" is an ambiguous word since this word is often used in different situations and with different roles in the sentence, such as: delimiter of *EDU* (should be classified as *B-EDU*); in combination with other words that are themselves delimiters, such as *"dado que"*, *"já que"*, *"uma vez que"* (should be classified as *O* and the previous word as *B-EDU*), and in the beginning of a parenthetical span of text (should be classified as *O* or *I-VP*).

| gold / pred | O | B-EDU | B-VP | I-VP |
|---|---|---|---|---|
| O | 10213 | 20 | 0 | 1672 |
| B-EDU | 26 | 2674 | 90 | 428 |
| B-VP | 0 | 39 | 6242 | 0 |
| I-VP | 367 | 206 | 0 | 33566 |

Table 4.3: EDU Boundary Detection Confusion Matrix

### 4.3.5 Conclusions

In this section we described the proposed approach to address the problem of ADU Boundary Detection, which corresponds to the second step of the argumentation mining process.

In Section 4.1, we considered the sentence level as the elementary unit under analysis. From a detailed analysis of the annotations contained in the ArgMine corpus, we concluded that a deeper analysis on smaller spans of texts should be formalized when addressing the problem of ADU Boundary Detection. The identification of the exact boundaries of the argumentative discourse units at word level is very challenging due to the high number of possible splitting points in a sentence and due to the fuzziness of the boundaries associated to the annotated ADU's. In order

to perform this task based on the ArgMine corpus, an higher quantity and quality of annotations would be required. Thus, we have used a two step approach to address the current task and the following hypothesis was considered: given an argumentative sentence, ADU boundaries corresponds to EDU boundaries, where EDU boundaries are defined as the convention used by the RST framework. The detection of argumentative sentences from free text was formulated and described in Section 4.1. Then, we rely on an external dataset annotated following the RST framework to train a sequential classifier in the task of dividing a sentence into smaller spans of text, which corresponds to EDUs defined in the RST framework. The overall results are good, in the task of EDU Boundary learning. From the hypothesis made in this section, we assume that the token level boundaries detected by a sequential classifier to perform EDU boundary detection in a corpus annotated following the RST framework, correspond to the ADUs boundaries if the corresponding sentence is argumentative. However, since the argumentation theory followed in the ArgMine corpus and the discourse theory followed in the CSTNews corpus have several conceptual differences and were defined for different purposes, we expect that the performance of the system when applied to ADU boundary detection may decrease comparing with the performance of the system when applied to EDU boundary detection. Evaluating the applicability of the classifier developed in this section to perform ADU boundary detection in the ArgMine corpus was out of scope for this thesis and should be properly addressed in future work.

The predictive capabilities of the model developed in this section are being applied to suggest, in argumentative sentences, the exact boundaries of each ADU in the *Annotation Platform* (Section 3.1).

# Chapter 5

# Conclusions

Argumentation mining is a growing research topic that spans across diverse research areas such as artificial intelligence, natural language processing, philosophy and computer science. Initial studies started to appear only few years ago and within specific genres. Since 2014, in which at least three international events on argumentation mining were organized, this research topic is gaining visibility at major artificial intelligence and computational linguistic conferences, and is gaining attention from major corporations. For instance, IBM has recently funded a multimillion cognitive computing project in which argumentation mining is the core technology[1]. Argumentation mining is not only an engaging problem, but also a research topic with potential for several applications in different domains such as debates, multi-agent systems, processing of user-generated content, and so on.

The aim of argumentation mining from text, a sub-domain of text mining, is the automatic detection and identification of the argumentative structure contained within a piece of natural language text. As input, this process receives a piece of natural language text. As output, this process aims to represent the structure of the arguments presented in the text document with the corresponding argument diagram. There are some characteristics of natural language text and from the argumentation process that make argumentation mining a very challenging task, such as the ambiguity of natural language text, different writing styles, implicit context and the complexity of building argument structures.

Typical approaches to argumentation mining follow the supervised machine learning paradigm, in which a set of labeled data (corpus) is necessary in order to build a model that learns how to map inputs to the desired outputs. In this thesis, we studied argumentation mining applied to texts written in the Portuguese language. Therefore, a corpus containing texts annotated with arguments in the Portuguese language is required. Since, to the best of our knowledge, no such corpus existed, we created an annotation platform in order to start the process of creation of an annotated corpus with arguments from texts written in the Portuguese language.

[1] http://researcher.watson.ibm.com/researcher/view_group.php?id=5443

The full task of argumentation mining can be decomposed in several subtasks, as described in Section 2.3, namely: segmentation, identification of argumentative discourse units, argumentative discourse units classification, relation identification and relation type classification.

In order to integrate the creation of a corpus containing news articles annotated with arguments in the Portuguese language and the semi-automated process of selection and experimentation of different models and relevant features for different subtasks of the argumentation mining process, the *ArgMine Framework* is proposed.

In this thesis, we addressed the first two subtasks of the argumentation mining process, namely *Argumentative Sentence Detection* and *ADU Boundary Detection*.

To address the task of *Argumentative Sentence Detection*, semi-supervised machine learning techniques were applied and, a binary classifier was trained to detect argumentative sentences from free text, based on the annotations available in the *ArgMine Corpus*. From a detailed analysis of the obtained results, we conclude that the modest quantity of annotations available in the *ArgMine Corpus* and, in addition, the variability of topics that are covered in different articles make the current task even more challenge to be addressed using machine learning algorithms. Besides increasing the quantity and quality of the annotations available in the *ArgMine Corpus*, it would be also interesting to improve the set of features currently used to address this task taking into consideration the characteristics of the arguments presented in the text and the variability of topics that are covered in the *ArgMine Corpus*. The classifier obtained from this section is used in the *Annotation Platform* (described in Section 3.1) to detect the zones of text that contain argumentative content.

To address the task of *ADU Boundary Detection* we made the assumption that given an argumentative sentence, the ADU's boundaries correspond to EDU's boundaries, where EDU boundaries are defined as the convention used by the RST framework. Then, a two step approach is used to address this task. In the first step, the detection of argumentative sentences from free text is determined using the model obtained from the previously described subtask, *Argumentative Sentence Detection*. In the second step, supervised machine learning techniques were applied, and a sequential classifier is trained using an external resource, CSTNews corpus, to identify the boundaries of the EDU's. The classifier obtained from this section is used in the *Annotation Platform* (described in Section 3.1) to identify the different components of the arguments contained in argumentative sentences.

## 5.1 Lessons Learned

In this section we would like to describe some of the major difficulties that we had to overcome during the development of this thesis.

Being argumentation mining a recent research topic, not only the concepts but also the approaches and targets vary widely between researchers. Some researchers use different argumentation models as the basis of their work, different definitions of arguments and their components, and different assumptions are made in relation to some subtasks of the argumentation mining process.

All these characteristics of a growing research topic hamper the interpretation and comparison of different research works and difficult the process of conceptualization and definition of the several tasks that must be addressed in order to successfully approach argumentation mining.

The aim of this thesis is to study argumentation mining from text documents written in Portuguese using supervised and semi-supervised machine learning algorithms. Due to the lack of a corpus annotated with arguments from texts written in Portuguese, we had to face the challenging task of initiating the process of creation of an annotated corpus. Firstly, it was necessary to formulate the argumentation mining process taking into consideration the target text documents that were available and the goals that we desire to achieve. This involves choosing the argumentation model that would be the base of our work, to properly define the different concepts involved in the topic of argumentation mining (e.g What is an argument? What is the granularity and boundaries of the argumentative discourse units that we aim to address? What argument structures should be considered? Should we consider implicit arguments?, and so on), creating guidelines that should be followed by annotators in the process of annotation and, finally, the creation of an annotation platform to facilitate the access to news articles written in Portuguese and to make the complex process of annotation of arguments from text more intuitive.

The skill needed for extracting the argumentative structure contained in text documents is complex, time consuming, requires training and good interpretation skills. It involves the detection, extraction and identification of the arguments and corresponding components being presented in the text document. Being involved in the process of annotation gave us important insights to address some of the subtasks in the argumentation mining process. We firmly believe that, the effort required to annotate arguments from text documents should not be underestimated.

Additional problems related to the processing of natural language text in Portuguese language were unexpected and difficult to solve. Some of the approaches that we would like to experiment when addressing some subtasks related to argumentation mining were difficult to be addressed because additional problems have to be tackled (e.g. named entity disambiguation). Improvements in some natural language processing problems for the Portuguese language would be highly beneficial to some approaches in the area of argumentation mining (e.g named entity disambiguation, part-of-speech taggers, dependency parsers, semantic parsers, amongst others). Combining different natural language processing tools, machine learning libraries and argumentation tools was also a challenging task because different tools have different interfaces and require different input/output formats. In order to integrate different tools and to facilitate future development and research in argumentation mining motivated the creation of the ArgMine Framework. All these challenges helped us to improve our critical analysis skills and, to better understand some of the phases involved in a machine learning system such as, feature engineering, data preparation, data understanding, amongst others.

Finally, applying machine learning techniques to address some of the subtasks of argumentation mining is very challenging due to the complexity of the task. Additionally, we had to face the problem of having a corpus with a reduced quantity of annotations and which was being created while we formulated the approach to some of the subtasks of the argumentation mining process.

## 5.2 Future Work

As defined in Section 2.3, the full argumentation mining process can be decomposed in several subtasks. In future work, we expect to integrate all the models obtained after addressing each subtask of the argumentation mining process, to automatically identify and extract the arguments contained in texts written in the Portuguese language. In this sense, we aim to continually use the *ArgMine Framework* when addressing each of the subtasks.

As described in this thesis, the *ArgMine Corpus* is at a beginning stage of its creation. In order to successfully address each of the subtasks in the argumentation mining process, several improvements should be made and several considerations should be taken into account. Firstly, the number of annotations contained in the *ArgMine Corpus* should be significantly higher. In order to successfully apply machine learning algorithms to a complex task, such as argumentation mining, in which the clues that are provided in the text often require to take combinations of several variables into consideration (for instance, lexical clues that are typically associated to argumentative content are often ambiguous, requiring syntactic and/or semantic information), the quantity and quality of annotations is crucial in order to make the learning process feasible. Secondly, the process of annotation of arguments from text have some characteristics that should be consider. It is necessary to implement more sophisticated methods (comparing to the methods applied in this thesis) to address the fuzziness of the boundaries in the annotated arguments: annotators tend to identify essentially the same units, but the boundaries differ slightly. This step has special relevance when transforming annotations into training instances. The fuzziness of the boundaries in the annotated arguments can difficult the identification of the elementary units that have to be defined in every step of the argumentation mining process. Improvements in this topic can have high impact in several problems of the argumentation mining process, can improve the consistency of the obtained training instances and is crucial to implement agreement metrics suited for argumentation mining. Thirdly, it is necessary to formulate inter-annotator agreement metrics (IAA) suited for the particularities of the argumentation mining process. Some improvements to commonly used IAA metrics that are more suited for argumentation mining have already been explored [WMGA14], and they are important to measure the quality and confidence of the annotated corpus.

Some of the topics that we think that would be relevant to improve the work presented in this thesis and that can have high impact to improve the current state-of-the-art in the research area of argumentation mining include:

- textual entailment: in the absence of clues directly indicating the argumentative structure contained in a text document, humans rely on context, domain and/or common-sense knowledge to understand and identify the arguments being presented. Textual entailment studies pair of words that when considering one of them as true people usually believe that the second one is also true. This approach seems consistent with the notion of steps of reasoning from the premises to the conclusion (i.e. conclusion follows from one or more reasoning steps from the premises) associated to the definition of argument and, therefore, seems a concept interesting to explore in argumentation mining;

- event causality: recognizing event causality is an important part of text understanding. Humans are good at inferring causal relations in a discourse and it is an important component on our world knowledge. Current work developed in this line focuses on methods of distributional semantics, such as co-occurrences counts of events collected automatically from an unlabeled corpus. This research topic is relevant to deal with the problem of common-sense presented in text documents;

- sentiment polarity: arguments are justifiable positions, in which, the conclusion is often presented as an opinion/position justified with facts or evidence. Therefore, sentiment polarity can be used to detect the spans of text where a position is being made;

- factoids: in an argument, premises correspond to the evidences or facts that are presented to support the conclusion that is being made. In this sense, factoids can be interesting to explore in order to detect premises from text;

- distributional semantics: in this thesis we explored the word embeddings representation at the word level to implement the feature *Domain words repetition* in the task *Argumentative Sentence Detection*. However, studying the embeddings representation at sentence or document level would be interesting to explore. Some researchers, such as Habernal and Gurevych [HG15], have already explored this direction, using the word embeddings representation at sentence level and reporting promising results;

- deep learning algorithms: researchers in natural language processing and machine learning reported outstanding improvements in previously addressed tasks when applying deep learning algorithms (e.g. Recurrent Neural Networks algorithm). Exploring this techniques in some subtasks of the argumentation mining process would be interesting to explore;

- levels of abstraction: in some arguments premises and conclusions are at different levels of abstraction (e.g. *"All men are mortal and Socrates is a man. Therefore, Socrates is mortal."*), which is used to express the relation of support or conflict between conclusion and premises. Studying techniques to detect changes in the level of abstraction between propositions may be an interesting research topic to explore and could be applied in some subtasks of the argumentation mining process;

- name entity disambiguation: transforming "implicit pointers" in the original text with "explicit pointers" can help machine learning approaches to find patterns in data and can improve the quality of the feature *Domain words repetition* implemented in the task *Argumentative Sentence Detection*;

- argument schemes: can play an important role to handle hidden assumptions (enthymemes). Typically, arguments that we encounter in a text document are arguments where some important assumptions are not made explicit. Writers omit some informations or part of the

arguments because they rely in the common sense and knowledge about the world that human readers possess. Comparing argumentation schemes with the argument presented in the text, the missing parts can be easily derived;

- statistical relational learning: aims to combine firs-order logic with statistical machine learning. The expressive power of first-order logic can be exploited to model background knowledge of a given domain and to represent the relations of support or attack between premises and conclusions. Statistical machine learning can be used to find patterns in data and, can naturally deal with uncertainty. From the definition of argument and, since we aim to detect and extract arguments from text, this research topic seems promising for argumentation mining.

# References


[AP94] J. J. Almeida and Ulisses Pinto. Jspell–um módulo para análise léxica genérica de linguagem natural. *Actas do X Encontro da Associação Portuguesa de Linguística*, pages 1–15, 1994.

[ARPS13] Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[AS11] Safia Abbas and Hajime Sawamura. ALES: an innovative agent-based learning environment to teach argumentation. *KES Journal*, 15(1):25–41, 2011.

[Aza99] Moshe Azar. Argumentative text as rhetorical structure: An application of rhetorical structure theory. *Argumentation*, 13(1):97–114, 1999.

[BD10] Bal Krishna Bal and Patrick Saint Dizier. Towards building annotated resources for analyzing opinions and argumentation in news editorials. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. European Language Resources Association (ELRA).

[Bn15] F. Boltužić and J. Šnajder. Identifying prominent arguments in online debates using semantic textual similarity. In *Proceedings of the 2nd Workshop on Argumentation Mining (ArgMining 2015)*, NAACL 2015, pages 110–115, Denver, Colorado, 2015. Association for Computational Linguistics.

[CM01] Lynn Carlson and Daniel Marcu. Discourse tagging reference manual. *ISI Technical Report ISI-TR-545*, 54, 2001.

[CMG09] Carlos Iván Chesñevar, Ana Gabriela Maguitman, and María Paula González. *Argumentation in Artificial Intelligence*, chapter Empowering Recommendation Technologies Through Argumentation, pages 403–422. Springer US, Boston, MA, 2009.

[CMJ+11] Paula C.F. Cardoso, Erick G. Maziero, María Lucía Castro Jorge, Eloize R.M. Seno, Ariani Di Felippo, Lucia Helena Machado Rino, Maria das Gracas Volpe Nunes, and Thiago Alexandre Salgueiro Pardo. Cstnews - a discourse-annotated corpus for single and multi-document summarization of news texts in brazilian portuguese. In *Proceedings of the 3rd RST Brazilian Meeting*, pages 88–105, Cuiabá, Brazil, 2011.

[CMM+06] C. Chesñevar, J. McGinnis, S. Modgil, I. Rahwan, C. Reed, G. Simari, M. South, G. Vreeswijk, and S. Wilmott. Towards an Argument Interchange Format. *The Knowledge Engineering Review*, 21(4):293–316, 2006.

[Coh84] Robin Cohen. A computational theory of the function of clue words in argument understanding. In *Proceedings of the 10th International Conference on Computational Linguistics*, COLING '84, pages 251–258, Stroudsburg, PA, USA, 1984. Association for Computational Linguistics.

[Coh87] Robin Cohen. Analyzing the structure of argumentative discourse. *Comput. Linguist.*, 13(1-2):11–24, January 1987.

[CV12] Elena Cabrio and Serena Villata. Natural language arguments: A combined approach. In *ECAI*, volume 242, pages 205–210, 2012.

[Dav12] Martin Davies. Computer-aided mapping and the teaching of critical thinking. *Inquiry: Critical Thinking Across the Disciplines*, 27(2):15–30, 2012.

[Dun95] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–357, September 1995.

[Fre91] James B. Freeman. *Dialectics and the Macrostructure of Arguments: A Theory of Argument Structure*. Foris Publications, 1991.

[GG15] Marcos Garcia and Pablo Gamallo. Yet another suite of multilingual NLP tools. In José Paulo Leal José-Luis Sierra-Rodríguez and Alberto Simões, editors, *Languages, Applications and Technologies. Communications in Computer and Information Science*, volume 563, pages 65–75. Springer, 2015.

[GLPK14] Theodosios Goudas, Christos Louizos, Georgios Petasis, and Vangelis Karkaletsis. Argument extraction from news, blogs, and social media. In *Artificial Intelligence: Methods and Applications - 8th Hellenic Conference on AI, SETN 2014, Ioannina, Greece, May 15-17, 2014. Proceedings*, pages 287–299, 2014.

[Gre10] Nancy L Green. Representation of argumentation in text with rhetorical structure theory. *Argumentation*, 24(2):181–196, 2010.

[HEKG14] Ivan Habernal, Judith Eckle-Kohler, and Iryna Gurevych. Argumentation mining on the web from information seeking perspective. In Elena Cabrio, Serena Villata, and Adam Wyner, editors, *Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing*, pages 26–39. CEUR-WS, July 2014.

[HG15] Ivan Habernal and Iryna Gurevych. Exploiting debate portals for semi-supervised argumentation mining in user-generated web discourse. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2127–2137, Lisbon, Portugal, 2015. Association for Computational Linguistics.

[JLR14] Mathilde Janier, John Lawrence, and Chris Reed. OVA+: an argument analysis interface. In *Computational Models of Argument - Proceedings of COMMA 2014, Atholl Palace Hotel, Scottish Highlands, UK, September 9-12, 2014*, pages 463–464, 2014.

[LMP01] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML

'01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[LR14] John Lawrence and Chris Reed. Aifdb corpora. In *Computational Models of Argument - Proceedings of COMMA 2014, Atholl Palace Hotel, Scottish Highlands, UK, September 9-12, 2014*, pages 465–466, 2014.

[LR15] J. Lawrence and C. Reed. Combining argument mining techniques. In *Proceedings of the 2nd Workshop on Argumentation Mining*, page 127–136. Association for Computational Linguistics, 2015.

[LRA+14] John Lawrence, Chris Reed, Colin Allen, Simon McAlister, and Andrew Ravenscroft. Mining arguments from 19th century philosophical texts using topic based modelling. In *Proceedings of the First Workshop on Argumentation Mining*, pages 79–87, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[LT16] Marco Lippi and Paolo Torroni. Argumentation mining: State of the art and emerging trends. *ACM Trans. Internet Technol.*, 16(2):10:1–10:25, March 2016.

[Mar00] Daniel Marcu. The rhetorical parsing of unrestricted texts: A surface-based approach. *Computational linguistics*, 26(3):395–448, 2000.

[MBPR07] Marie-Francine Moens, Erik Boiy, Raquel Mochales Palau, and Chris Reed. Automatic detection of arguments in legal texts. In *Proceedings of the 11th International Conference on Artificial Intelligence and Law*, ICAIL '07, pages 225–230, New York, NY, USA, 2007. ACM.

[MI09] Raquel Mochales and Aagje Ieven. Creating an argumentation corpus: do theories apply to real arguments?: a case study on the legal argumentation of the echr. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 21–30. ACM, 2009.

[Mit97] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[MP11] Raquel Mochales Palau. *Automatic Detection and Classification of Argumentation in a Legal Case*. PhD thesis, Faculty of Engineering Science, July 2011. Moens, Marie-Francine and De Schreye, Daniel (supervisors).

[MT88] William C Mann and Sandra A Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281, 1988.

[PC14] Joonsuk Park and Claire Cardie. Identifying appropriate support for propositions in online user comments. In *Proceedings of the First Workshop on Argumentation Mining*, pages 29–38, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[PM09] Raquel Mochales Palau and Marie-Francine Moens. Argumentation mining: The detection, classification and structure of arguments in text. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, ICAIL '09, pages 98–107, New York, NY, USA, 2009. ACM.

[PS13] Andreas Peldszus and Manfred Stede. From argument diagrams to argumentation mining in texts: A survey. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 7(1):1–31, 2013.

[PS15] Andreas Peldszus and Manfred Stede. Joint prediction in mst-style discourse parsing for argumentation mining. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 938–948, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[RCT16] Gil Rocha, Henrique Lopes Cardoso, and Jorge Teixeira. ArgMine: A Framework for Argumentation Mining. In *Computational Processing of the Portuguese Language - 12th International Conference, PROPOR 2016, Student Research Workshop, Tomar, Portugal, July 13-15*, 2016.

[Ree06] Chris Reed. Preliminary results from an argument corpus. In *In Eloína Miyares Bermúdez & Leonel Ruiz Miyares (Eds), Linguistics in the twenty-first century*, pages 185–196. Scholars Press, 2006.

[RM12] Sara Rosenthal and Kathleen McKeown. Detecting opinionated claims in online discussions. In *Proceedings of the 2012 IEEE Sixth International Conference on Semantic Computing*, ICSC '12, pages 30–37, Washington, DC, USA, 2012. IEEE Computer Society.

[RWB12] Niall Rooney, Hui Wang, and Fiona Browne. Applying kernel methods to argumentation mining. In *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference, Marco Island, Florida. May 23-25, 2012*, 2012.

[RWM07] C. Reed, D. Walton, and F. Macagno. Argument diagramming in logic, law and artificial intelligence. *Knowl. Eng. Rev.*, 22(1):87–109, March 2007.

[Sai12] Patrick Saint-Dizier. Processing natural language arguments with the TextCoop platform. *Argument & Computation*, 3(1):49–82, 2012.

[SG14] Christian Stab and Iryna Gurevych. Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 46–56, 2014.

[SM12] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Found. Trends Mach. Learn.*, 4(4):267–373, April 2012.

[SW12] Jodi Schneider and Adam Wyner. Identifying consumers' arguments in text. In *Proceedings of the 1st Workshop on Semantic Web and Information Extraction (SWAIE 2012)*, Galway, Ireland, 2012.

[TM02] Simone Teufel and Marc Moens. Summarizing scientific articles: Experiments with relevance and rhetorical status. *Comput. Linguist.*, 28(4):409–445, December 2002.

REFERENCES


[Tou58] S.E. Toulmin. *The Uses of Argument*. Cambridge University Press, 1958.

[vE01] F.H. van Eemeren. *Crucial Concepts in Argumentation Theory*. Amsterdam University Press, 2001.

[vEG04] F.H. van Eemeren and R. Grootendorst. *A Systematic Theory of Argumentation: The Pragma-dialectical Approach*. Ariel Empresa. Cambridge University Press, 2004.

[Wal96] D.N. Walton. *Argumentation Schemes for Presumptive Reasoning*. Argumentation Schemes for Presumptive Reasoning. L. Erlbaum Associates, 1996.

[Wha36] Richard Whately. *Elements of Logic*. 6th ed. London: B. Fellowes, 1836.

[WMGA14] Nina Wacholder, Smaranda Muresan, Debanjan Ghosh, and Mark Aakhus. Annotating multiparty discourse: Challenges for agreement metrics. In *Proceedings of LAW VIII - The 8th Linguistic Annotation Workshop*, pages 120–128, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University.